Author:
O'Connor, Michael

Title:
Accelerated Sampling Methods for High Dimensional Molecular Systems

Accelerated Sampling Methods for High Dimensional Molecular Systems

By

MICHAEL O'CONNOR

Department of Computer Science
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

JANUARY 2018

Word count: 46264

Abstract

Simulating molecular systems is a significant use of high-performance computing. However, molecular dynamics simulations are limited by the time scales of events that can be reliably accessed. Such events are often precisely those that are of interest, such as chemical reactions or substantial conformational changes. Accelerated simulation methods can bias a simulation towards these so-called 'rare events' more quickly, but they are typically computationally expensive or require the identification of low-dimensional representations that describe the event.

In this thesis, improvements to simulation methods for accessing these rare events are presented. The first is the automation and generalisation of the boxed molecular dynamics (BXD) method, making it more efficient and usable with events that require higher-dimensional representation. Additionally, a virtual reality framework for interactive molecular dynamics (iMD-VR) is presented as a strategy for the rapid identification of pathways and collective variables that can be used with existing accelerated molecular dynamics methods.

The framework is evaluated in a user study, in which it was found that VR enables a statistically significant advantage over traditional interfaces for performing tasks in molecular systems. Furthermore, the framework is evaluated for generating initial pathways on a benchmark system, alanine dipeptide, and found to produce reasonable pathways. These pathways were then optimised for use with metadynamics, an accelerated sampling method, to produce converged free energy surfaces. The framework is further tested in larger systems including knotting pathways in the hypothetical protein MJ0366 and loop motions in the enzyme cyclophilin A. In these cases, it was possible to produce the desired pathways and initial conditions with the iMD-VR framework. The use of adaptive sampling with Markov models to perform follow-up sampling on these systems is critically evaluated.

DEDICATION AND ACKNOWLEDGEMENTS

This thesis is dedicated to the community of scientists, hardware engineers, and open-source software engineers whose achievements make the contributions of this dissertation possible. In that spirit, there are many people to thank who have contributed in some way.

Firstly, my sincere thanks to my supervisors David R. Glowacki and Simon McIntosh-Smith for their guidance, enthusiasm and expertise, and for always providing everything needed to conduct my research in the way of funding, travel, equipment and ideas.

Additionally, I wish to thank everybody at Interactive Scientific Ltd for supporting my studies through a CASE studentship. In particular I wish to thank Becky Sage, Phill Tew, Balazs Hornung and Mark Wonnacott, from whom I have learnt so much.

I also thank everyone with whom I have crossed paths in my time at the Centre for Computational Chemistry, for making me feel welcome in a new field, for the fruitful discussions over the many (and often long) tea breaks, and for their enthusiasm for collaborations in which we were able to share expertise across a wide variety of domains. I would like to specially thank Robert Arbon, Simon Bennie, Helen Deeks, Alex Jameson-Binnie, Alex Jones, Professor Adrian Mulholland, Robin Shannon, Lisa May-Thomas and Rebecca Walters for their contributions to the project. I also thank Jordi Juarez Jimenez, at the University of Edinburgh, for his input and expertise.

Finally, my heartfelt thanks to my wife Susanna, and my family, for supporting me along this long and winding journey.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

TABLE OF CONTENTS

List of Tables

Table **Page**

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

With the rapid advances in computational power over the latter half of the 20th century and the early 21st century, the simulation of molecular systems with computational models has become a widely-used tool for both basic science research and applied science, forming large portions of high-performance computing workloads[1, 2]. Unlike experimental research, simulated models of a molecular system can capture the entire state and dynamics of the system[3], providing minute detail into the processes at scales not visible to experiment. For the basic research scientist, this detail can provide insight into how molecular systems work, such as how a protein folds[4], the reaction mechanism of a particular enzyme[5], or the mechanism of a chemical reaction[6]. For scientists working in applications or industry, simulations can provide insight into material design[7], or provide an automated-screening process for new materials and drug candidates, lowering costs[8].

While improvements in algorithms, increasing compute power and the adoption of parallel computer architectures enables the study of large systems, such as simulations of an entire ribosome in an explicit water solvent, consisting of 2 million atoms[9], or 20 trillion atoms in a Lennard Jones fluid[10], the timescales a molecular simulation can access is limited. In a molecular dynamics simulation, a simulation propagates forward in time with the resulting state of the previous step affecting the next. Molecular simulations are thus limited by how quickly a single step can be computed, and the size, in units of time, of this step. The small vibrations of a molecular system occur on the timescale of femtoseconds, while the process of interest, such as a chemical reaction or

a transition to a new configuration, may occur much more slowly, on the timescales of milliseconds or seconds. Balancing the complexity of the model, the size of the system and the event to be studied such that it can be simulated within a reasonable time on the available compute power is a constant concern for the computational chemist.

This thesis is concerned with the problem of optimising simulations to access these so-called 'rare events'. Many strategies for this problem have been proposed and are in use, all of which have advantages and disadvantages. This thesis identifies some of the limitations of these methods, and through the development of new algorithms that are appropriate for modern computer architectures, and the use of novel human-computer interaction methods, develops and evaluates new methodologies and workflows that can be combined with existing methods to enable computational chemists to more rapidly identify and sample novel events.

1.1 Thesis Outline

This thesis is organised into the following chapters:

- In Chapter 2 an overview of the principles of molecular dynamics is given to provide context on the rare event problem, and a review of the approaches used to tackle rare event sampling is described. The successes and shortcomings of these existing methods are discussed, laying the foundations on which the contributions of this thesis are built.

- In Chapter 3 the boxed molecular dynamics algorithm (BXD) is discussed in detail, and extensions to the algorithm are presented. These include an automated procedure for generating boundaries and a generalisation to multidimensional collective variable space. An application of the extensions made to the BXD algorithm is presented in the accelerated sampling of the abstraction of deuterium from acetonitrile in both the solution and gas phases.

- Chapter 4 describes a platform developed for performing interactive molecular dynamics using virtual reality (iMD-VR) and cloud computing. The performance of the platform is evaluated in the context of simulation size, rendering and network performance.

- In Chapter 5, a user study performed to evaluate the utility of virtual reality in performing complex molecular tasks is presented and evaluated in the context of accelerated sampling.

- In Chapter 6 algorithmic strategies for analysing the output of the iMD-VR platform is evaluated through combination with existing methods to accelerate sampling of pathways in alanine dipeptide.

- In Chapter 7 the utility of the iMD-VR platform is evaluated further in the formation of slipknots in the protein MJ0366 and the accelerated sampling of loop motions in cyclophilin A, through combination with Markov State modelling.

- In Chapter 8 the developments of the previous chapters are discussed in order to summarise the findings and give an outlook on future work.

1.2 Contributions

Due to the highly collaborative nature of some of the work undertaken in this thesis, the major contributions, and their contributors, are listed in detail below:

- A novel automation procedure for the boxed molecular dynamics algorithm is introduced, making it easier, less error-prone and more efficient to use. By adaptively generating BXD boundaries on the fly during a molecular dynamics run, the overhead in setting up accelerated sampling along a reaction coordinate is reduced. David Glowacki and I designed the method, and I developed, implemented and evaluated the algorithms, with input from supervisors David Glowacki and Simon McIntosh Smith.

- A generalisation of the BXD algorithm to multidimensional collective variable space. The generalisation allows BXD to be used in a wider variety of molecular systems, where it is often the case that two or more collective variables are required to describe a process. Furthermore, a general velocity inversion procedure in collective variable space is developed, which while a key part of the BXD algorithm, may also have uses in other areas of molecular dynamics. I designed, developed, implemented and evaluated the algorithms, with input from supervisors David Glowacki and Simon McIntosh Smith.

- An open-source platform for performing interactive molecular dynamics (iMD-VR) using virtual reality is developed. The software was developed in collaboration with developers at Interactive Scientific Ltd, and developers at the University of Bristol building on previous molecular dynamics implementations of Glowacki and co-workers[11]. Rebecca Sage, Phill Tew, Mark Wonnacott and I developed much of the original software together, with Rebecca Sage managing the project, Phill Tew developing the transport layer, the cloud architecture and providing guidance on the overall software design and contributing to the virtual reality application, and with Mark Wonnacott developing the initial front-end implementation for tablets and smartphones, and developing the molecular renderers. My key contributions were contributing to the design and development of software architecture and algorithms. These included algorithms for applying bias potentials to molecular dynamics simulations interactively, the initial virtual reality application, software for a user study that evaluated the utility of virtual reality for 3D molecular manipulation, and a modular framework allowing the platform to be used with a wide variety of packages. I additionally manage the open-source project. Researcher Simon Bennie and I developed the DFTB+ plugin, and Simon Bennie developed the DL-POLY integration. Researcher Helen Deeks designed and performed the user study, for which I developed software and provided technical support, detailed in Chapter 5. The project was managed by David R. Glowacki, who provided support, guidance and ideas.

  Since its release as open-source software, researchers Helen Deeks, Alex Jameson-Binnie, Alex Jones, and Rebecca Walters have made significant contributions and improvements, under my guidance and direction.

- The combination of the iMD-VR platform with existing rare event methods to enable accelerated sampling of a variety of molecular systems, and subsequent evaluation of the methods. These include using iMD-VR to generate a path for use with the path collective variable and metadynamics, and as initial seeds for combination with adaptive sampling using Markov State Models. I developed the necessary methods and software, ran the experiments and performed the analysis of results. Researcher Jordi Juarez Jimenez from the University of Edinburgh provided the structure and original input files used in the study of cyclophilin A in Chapter 7, and guidance on the loop motion that was studied.

These contributions have led to the following publications:

- M. O'Connor, P. Tew, B. Sage, S. McIntosh-Smith and D. R. Glowacki, "Nano Simbox: An OpenCL-accelerated Framework for Interactive Molecular Dynamics" in Proceedings of the 3rd International Workshop on OpenCL, ACM, 2015, p. 20:1–20:1.

  Summary of Contributions: MO and PT designed and implemented the application. BS, SMS and DRG managed the overall project. MO wrote the initial draft with subsequent input from DRG.

- M. O'Connor, E. Paci, S. McIntosh-Smith, and D. R. Glowacki, "Adaptive boxed molecular dynamics in multidimensional collective variable space," Faraday Discussions (2016), doi:10.1039/C6FD00138F.

  Summary of Contributions: MO and DRG designed the algorithms. MO implemented the algorithms and performed the computational experiments, and wrote the initial paper draft along with DRG, with subsequent input from EP and SMS.

- M. O'Connor, H. M. Deeks, E. Dawn, O. Metatla, A. Roudaut, M. Sutton, B. R. Glowacki, L. M. Thomas, R. Sage, P. Tew, M. Wonnacott, P. Bates, A. J. Mulholland, D. R. Glowacki, "Sampling molecular conformations and dynamics in a multi-user virtual reality framework", 2018, Science Advances 4(6), doi:10.1126/sciadv.aat2731.

  Summary of Contributions: MO, PT, MW, and RS designed and implemented the cross-platform, real-time, cloud-mounted multiperson IMD framework. HMD and ED carried out user studies and performed data analysis. HMD, OM, and AR designed the user studies. MS constructed the video figures. BRG and DRG designed the molecular tasks in Fig. 2. AJM and DRG conceived the TEM-1 b-lactamase/benzylpenicillin application, which was developed by HMD under their supervision. PB provided crucial support in implementing the cloud-mounted simulation infrastructure. DRG designed the overall project concept, organised execution of the work strands, analysed the data, and wrote the initial paper draft along with HMD and MO, with subsequent input from AJM, LMT, BRG, OM, and AR

- R. J. Shannon, S. Amabilino, M. O'Connor, D. V. Shalishilin, D. R. Glowacki, "Adaptively Accelerating Reactive Molecular Dynamics Using Boxed Molecular Dynamics in Energy Space", Journal of Chemical Theory and Computation, 14(9), 4541–4552. doi:10.1021/acs.jctc.8b00515.

Summary of Contributions: RJS and SA designed and implemented the BXDE algorithm, performed the computational experiments and analysed the results. RS derived the required mathematical framework, with input from MO, and RJS, SA, MO and DRG wrote the initial paper draft with subsequent input from DS.

- M. O'Connor, S.J. Bennie, H.M. Deeks, A. Jamieson-Binnie, A.J. Jones, R.J. Shannon, R. Walters, T. Mitchell, A.J. Mulholland, D.R. Glowacki, "An open-source multi-person virtual reality framework for interactive molecular dynamics: from quantum chemistry to drug binding", arXiv:1902.01827, Journal of Chemical Physics, *submitted*, 2019.

  Summary of Contributions: DRG, SJB, HMD, AJB, AJ, and MO developed the necessary software. SJB performed the zeolite experiments, HMD and BW performed the ligand binding simulations, with supervision from AJM, BRG and AJB developed the data glove prototypes, AJ developed the sonification algorithms, RJS and MO developed the chemical reaction discovery prototype, and MO performed the cyclophilin A study. DRG wrote the initial paper draft, with input from all authors.

- S. J. Bennie, K. Ranaghan, H. M. Deeks, H. Goldsmith, M. O'Connor, A. J. Mulholland, D. R. Glowacki, "Teaching Enzyme Catalysis Using an Open Source Framework for Interactive Molecular Dynamics in Virtual Reality", ChemRxiv, doi: 10.26434/chemrxiv.7819982.v1, 2019.

- H. M. Deeks, R. K. Walters, M. O'Connor, A. J. Mulholland and D. R. Glowacki, "Fully flexible ligand docking to proteins using interactive molecular dynamics in virtual reality", *in preparation*.

CHAPTER 2

MOLECULAR DYNAMICS

2.1 Molecular Dynamics

The study of molecular systems using theoretical models was one of the early uses of computers for basic science research[12]. The use of computers arose out of the need for fast and accurate numerical computation[13]. While the properties of a single particle and the interaction of two particles could be described analytically, the behaviour of many interacting particles cannot be dealt with exactly. Even a small extension from two particles to three represents a huge difficulty for analytical methods, and before the advent of computers, researchers calculated trajectories of simple chemical reactions by hand[14]. The development of computers paved the way for studying molecular systems computationally rather than analytically. The landmark use was the development of the Monte Carlo methods in the 1950s, and its use to study small systems of hard spheres[12, 15, 16]. This was followed by the first molecular dynamics simulations of liquid argon using a Lennard-Jones potential[13], eventually leading to simulations of more complex liquids such as the first study of liquid water[17]. Molecular dynamics is distinct from the Monte Carlo method in that it propagates the molecular system through Newtonian mechanics, meaning that both static and dynamic behaviours can be probed[13, 17].

With computational power accelerating at an exponential rate, it was not long before more complex molecular systems could be studied. Using empirical potential energy functions and their derivatives developed by Warshel and Levitt[18–21], in a landmark

study Karplus *et al.* performed the first molecular dynamics trajectory of a protein[22]. While the potentials used were inaccurate by today's standards[23], and the trajectory a mere 9.2 picoseconds, the demonstration of the protein fluctuating around the structure observed with X-ray crystallography highlighted the importance of dynamics as it relates to biochemical structure and function. Since then, improvements in methodology, models and computation make molecular dynamics a valuable tool in diverse fields such as atmospheric chemistry[6], material science and drug discovery[8].

2.1.1 Theory

Classical molecular dynamics is a method for exploring configurations of molecular systems and computing properties of them, by propagating an initial state of the system forward in time under a set of equations of motion[24]. The bodies are the nuclei of the N atoms of the system. The interaction between the nuclei of the atoms are modelled with a potential energy function, V, which is related to the forces acting on the atoms via

$$\vec{f}(t) = -\nabla V(\vec{r}(t)), \tag{2.1}$$

where $\vec{r}(t) \in \mathbb{R}^{3N}$ is the vector consisting of the positions of each atom at time t and $\vec{f}(t) \in \mathbb{R}^{3N}$ is the resulting vector of forces acting on the atoms at time t.

By iteratively solving the differential equations in Newton's second law, the famous $F = ma$ (in one dimension), classical molecular dynamics uses the relation above to propagate the system according to the underlying potential energy function. For a system of N particles, this may be written as

$$\vec{a}(t) = \mathbf{M}^{-1}\vec{f}(t), \tag{2.2}$$

where $\mathbf{M} \in \mathbb{R}^{3N\times 3N}$ is the diagonal matrix of atomic masses, and $\vec{a} \in \mathbb{R}^{3N}$ is the vector of atomic accelerations. As an n-body problem, numerical methods are generally required to solve these equations for any system larger than two atoms[3]. In molecular dynamics, the most commonly used method is the finite difference method, typically in the form of verlet integration[3, 24]. In this method, the system is broken down into configurations separated by a small fixed amount of time denoted δt. The positions, velocities and forces at a particular time t are used to propagate the system forward to time $t+\delta t$,

which are then used to propagate the system on to time $t+2\delta t$ and so on. Velocity Verlet, a specific flavour of this integration technique, uses the following equations to propagate the positions and velocities:

$$\vec{r}(t+\delta t) = \vec{r}(t) + \delta t\vec{v}(t) + \tfrac{1}{2}\delta t^2\vec{a}(t), \tag{2.3a}$$

$$\vec{v}(t+\delta t) = \vec{v}(t) + \tfrac{1}{2}\delta t\,(\vec{a}(t) + \vec{a}(t+\delta t))\,. \tag{2.3b}$$

The distinction between molecular systems and other n-body problems is how the particles interact, which is introduced into the equations of motion via the forces. A wide range of models exist, generally trading accuracy for computational expense. The majority of practical methods make the Born-Oppenheimer approximation - that the nuclear positions can be considered fixed during the calculation of electronic movements. From this starting point, at the most accurate end are the *ab initio* methods, such as coupled-cluster theory[25], for solving the electronic structure (the motion of electrons) of a molecular system. Such methods can yield very accurate energies and properties of a molecular system, but are extremely computationally expensive and so are not generally practical for molecular dynamics[26].

At the other extreme, in the study of large systems at atomic detail, empirically fitted functions based on classical mechanics are used, in what is known as *molecular mechanics* (MM)[27]. Such methods are inexpensive compared to the *ab initio* methods, but do not model electronic structure so therefore typically cannot model chemical reactions.

In the spectrum between these two extremes, there many methods ranging from *ab initio* methods such as density functional theory (DFT), which, depending on system size, basis set choice and functionals used, can be used for molecular dynamics[28], to methods such as density functional theory tight-binding (DFTB)[29, 30] and semi-empirical methods such as PM6[31], which replace some of the more expensive calculations in electronic structure with empirically fitted functions or parameters. Such methods are able to model complex chemical reactions, while still being computationally cheap enough for molecular dynamics, for small molecular systems.

In the study of very large molecular systems, coarse-grained models which, as their names suggest, take a low-resolution approach by grouping atoms into units, such as amino acids in a protein, can be used[32].

It is also possible to combine models, with a specific small area of interest modelled with a more accurate model, and the surrounding system modelled with less accurate

models. The most popular of these is the QM/MM methods, combining an electronic structure method such as DFT with molecular mechanics[33–35].

In the systems that follow in this document, molecular mechanics force fields are used, but it should be noted that the principles, methods and algorithms apply equally to systems using other force fields. The choice of model is always a trade-off between the available computing power, the timescale of the event of interest, and the required accuracy.

Molecular Mechanics

In a classical molecular mechanics force field, the interaction of atoms is computed using pair-wise interatomic potentials which approximately describe molecular behaviour. For example, the van der Waals interaction of two atoms is often described using the Lennard-Jones approximation[36, 37]:

$$V_{VDW} = \sum_{i>j} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right],$$

where ϵ is the depth of the potential well, σ is the distance at which the potential is zero, and r_{ij} is the distance between the atoms i and j. As illustrated in Figure 2.1a, this potential results in repulsion at short range, attraction in the medium range, before decaying to zero at long distances.

The values for ϵ and σ may be fitted through comparison to experiment or an *ab initio* method. As well as the van der Waal interactions, the electrostatic interactions between atoms is typically approximated via Coulomb's law:

$$V_{elec} = \sum_{i>j} \frac{q_i q_j}{\epsilon r_{ij}},$$

where q_i and q_j are the partial charges of the atoms i and j, and ϵ is a dielectric constant. Such an atom-centric approach cannot easily model polarization, where the charge distribution of a molecule changes in response to its environment. Force fields that can model this, known as 'polarisable force fields', are thus an active area of development[38, 39].

The interaction of covalently bonded atoms is typically expressed as the sum of three components, the bond, angle and dihedral terms, and are depicted in Figure 2.1. The bond and angle terms typically take quadratic forms, which means they are essentially modelled as springs. The dihedral term may take a few forms, but a commonly used form is the cosine[40]:

$$V_{bonds} = \sum_{bonds} k_b(r - r_0)^2 \tag{2.4}$$

$$V_{angles} = \sum_{angles} k_\theta(\theta - \theta_0)^2 \tag{2.5}$$

$$V_{dihedrals} = \sum_{dihedrals} k_d(1 + \cos(n\phi - \phi_d)) \tag{2.6}$$

Here, k_b, k_θ and k_d are force constants governing the strength of the interaction, r is the distance between two bonded atoms, with r_0 the equilibrium bond distance, θ is the angle between two pairs of bonded atoms sharing an atom with θ_0 the equilibrium angle, ϕ is the dihedral angle of four connected atoms, n is the periodicity of the dihedral force and ϕ_d is the phase.

The total energy of the system in a classical molecular mechanics forcefield may hence be written as

$$V_{MM} = V_{VDW} + V_{elec} + V_{bonds} + V_{angles} + V_{dihedrals}$$

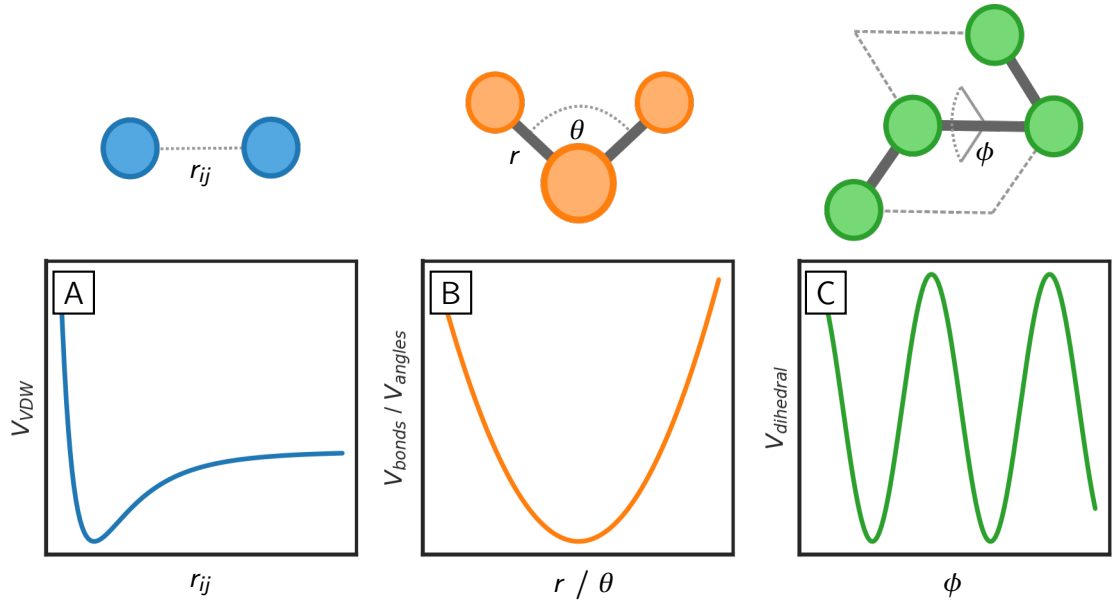


Figure 2.1: Illustrations of the potentials used in a molecular mechanics force field. A) The Lennard-Jones potential (V_{VDW}) for two particles i and j at a distance r_{ij}. B) The harmonic potential used for bonds between two atoms at a distance r and for modelling the angle, θ, between a triple of atoms. C) The potential used for modelling the dihedral angle, ϕ, between quadruples of bonded atoms. The pair of planes used to define the angle is shown as dashed lines parallel to the bonds between the relevant atoms.

The evaluation of the potential energy functions and their gradients make up the most expensive part of a molecular dynamics simulation, with the Lennard Jones and electrostatic potentials scaling $O(N^2)$ as written, and the bonded forces scaling $O(N)$. Through the use cell lists[41] the overall scaling can be reduced to $O(N)$. These force fields are the basis of all-atom molecular dynamics calculations for non-reactive systems.

An important observation to make is that as the bonds are modelled with predetermined pairwise quadratic expressions, the force between atoms increases quadratically at large distances, and so bonds cannot be broken. This means that molecular mechanics force fields such as those described here can only be used to study conformational changes, and cannot treat chemical reactions involving bond formation or breakage. To enable the study of such reactions, other force fields such as ReaxFF[42] or the Empirical Valence Bond methods[43–45] incorporate strategies for introducing reactive potentials into molecular mechanics.

2.1.2 Computing Observables and the Ergodic Hypothesis

Computational experiments using molecular dynamics simulations can be used to calculate many equilibrium and transport properties of a simulation[24]. Much like physical experiments, these properties are calculated by preparing the system appropriately, then running the simulation during which measurements are taken. The measured value of the property we are interested in, A, is then taken as the average of these measurements, $\langle A \rangle$.

The basis for this method of computing observables from molecular dynamics is the ergodic hypothesis, which states that the ensemble average of an observable, taken by measuring the observable from many independently prepared states, is equal to the long time average of an observable, taken by measuring the observable repeatedly over a long time[46]:

$$\langle A \rangle = Z^{-1} \int A(\vec{r},\vec{p}) e^{-H(\vec{r},\vec{p})/k_B T} \mathrm{d}\vec{r}\mathrm{d}\vec{p} = \lim_{\tau \to \inf} \tau^{-1} \int_0^\tau A(\vec{r}(t),\vec{p}(t))\mathrm{d}t. \tag{2.7}$$

Here, $\vec{p} \in \mathbb{R}^N$ is the vector of atomic momenta, $Z = \int e^{-H(\vec{r},\vec{p})/k_B T} \mathrm{d}\vec{r}\mathrm{d}\vec{p}$ is the partition function, k_B is the Boltzmann constant and T is the temperature of the system. $H(\vec{r},\vec{p})$ is the Hamiltonian describing the total energy of the system, typically written as

$$H(\vec{r},\vec{p}) = \sum_{i=1}^{N} \frac{||\vec{p}_i||^2}{2m_i} + V(\vec{r}),$$

where $||\vec{p}_i||^2$ is the square magnitude of the momentum of atom i and m_i is the mass of atom i.

The ergodic hypothesis means that in principle if one runs a molecular dynamics simulation for long enough, the average value of an observable will converge to the value it would have taken if we were able to generate initial conditions across the entirety of the relevant volume of phase space.

As we shall shortly see, in practice it is often not possible to run molecular dynamics trajectories for long enough in order to satisfy this ergodic hypothesis.

2.1.3 Reaction Coordinates, Collective Variables and Dimensionality Reduction

The phase space volume of any given molecular system is comprised of the $6N$-dimensional set of all possible values of the position and momentum variables, where N is the number of atoms, and represents all possible configurations of the system. With such high-dimensionality, it is almost always necessary to project the dimensionality of the system down onto some metric of interest. The metric used depends on the context and required properties.

Following the definitions of Abrams and Bussi[47], a collective variable (CV) is a function $S(\vec{r},\vec{p})$ of the $6N$-dimensional space of atomic positions and momenta mapping onto an M-dimensional CV space $\vec{s} \equiv \{s_i | i = 1,\ldots,M\}$, where $M \ll N$. This definition may be used to refer to a collective variable that consists of multiple components, or the combination of multiple one-dimensional collective variables.

Projecting a molecular simulation onto a few collective variables is highly useful for the computation of meaningful observables. For example, the free energy along a collective variable, $\Delta G(\vec{s})$, is given by

$$\Delta G(\vec{s}) = -k_B T \ln P(\vec{s}), \tag{2.8}$$

where $P(\vec{s})$ is the probability density function describing the distribution of $\vec{s}$, T is the temperature and k_B is the Boltzmann constant[46]. This thermodynamic calculation can provide insight into the favourability of a process, the free energy barriers along it indicating the energy available for work required (see Figure 2.2) for the process, as well as providing a means for calculating entropic effects.

These collective variables can take many forms and are dependent on the system being studied, but commonly used variables include the distance between subsets of

atoms, dihedral angles, the radius of gyration and combinations of them[47]. Specific collective variables may be developed for particular problems, such as finding ligand binding sites in proteins[48]. Another common metric is the distance between configurations, the root mean square deviation (RMSD)[49], given by:

$$\sqrt{\frac{1}{N}\sum_i ||\vec{r}_i - \vec{r}_i^0||^2},$$

where $\vec{r}$ is the positions of the atoms in the configuration, and $\vec{r}^0 \in \mathbb{R}^{3N}$ is the position of the atoms in the reference configuration. It is common to perform this calculation on only a subset of the atoms of the system, such as the backbone atoms in a protein.

A one-dimensional collective variable, where $M = 1$, used to represent progress along a reaction or conformational pathway is typically called a reaction coordinate. When discussing one-dimensional collective variables, the term reaction coordinate with denotation ρ will be used in the rest of this document.

Finding good low-dimensionality representations of molecular systems is a crucial step in many analysis and simulation methods, both for interpreting the data and for making computational methods scale more efficiently with system size.

2.1.4 Markov State Models

As computing resources have increased in power through increased parallelism, projects such as Folding@Home have produced huge datasets of many trajectories[50–52]. Additionally, purpose-built machines such as the ANTON have made millisecond long trajectories accessible[4, 53]. The analysis of these trajectories becomes a 'Big Data' problem, with the need for sophisticated tools. Over the last two decades, the Markov state model (MSM) analysis methods for molecular simulations have been developed. Many excellent review articles have been written on the theory, development and practical application of Markov models[54–57]. Here, a brief overview necessary for the applications undertaken in this thesis is given.

Conceptually, the MSM approach performs two key operations. The first is to discretise the data into N states, through dimensionality reduction and clustering. The second is to take this discretised set of states and estimate the probability of transitioning between each state, forming a transition matrix $\mathbf{T}(\tau) \in \mathbb{R}^{N\times N}$ where $\mathbf{T}_{ij}(\tau)$ gives the probability of transitioning from state i to state j after some *lag time* τ. This transition matrix forms the central data structure around which many observables can be computed, including kinetic rates and free energies[56].

Building a Markov Model

In the MSM framework, trajectories are projected onto some lower dimensional feature space, such as dihedral angles or contact distances between relevant parts of the system. The feature space may then be projected further onto a lower dimensional space through dimensionality reduction techniques such as Principal Component Analysis (PCA) or Time-Structure Based Independent Component Analysis (TICA)[54, 55]. The TICA method is particularly appropriate for molecular dynamics trajectories because it maximises the autocorrelation of the transformed coordinates, enabling it to describe the slow motions that are relevant in a molecular simulation[58, 59].

Once the trajectories have been projected onto some low dimensional space, they are discretised into microstates through some clustering algorithm such as K-Centering[60], grouping similar configurations together. These microstates form the states of the Markov state model. The choice of projection and discretisation approach can have a significant impact on the resulting Markov model. Chodera and Noe observe that a "challenge in all these approaches is that they propose a distance metric *a priori*. Because trajectory data is always limited in quantity by practical simulation times, it is essential that configurations that are highly similar by this metric are actually kinetically related"[55]. One has to find a projection (or metric) that sufficiently captures the relevant kinetic process. In many ways, this is an analogous situation to the identification of collective variables in other methods discussed in previous chapters. While TICA is an important method for identifying slow motions, one still typically needs to choose some distance metric first on which to apply it. Identifying robust representations and procedures for discretizing a Markov state model is still an ongoing research area, with new dimensionality reduction methods such as variational autoencoders being proposed[61]. Additionally, new approaches that do not require the assumption of Markovian dynamics on the discretized system, coined Projected Markov Models, offer a potential way forward[62].

To construct the Markov state model, an estimation of the transition probabilities between the microstates produced through discretisation is required. The simplest way to do this is to simply proceed through the trajectories in a stride, known as the lag time, τ, and count how many times a trajectory transitions from a microstate i to a microstate j. A count matrix $\mathbf{C}(\tau)$ can be formed, where $\mathbf{C}_{ij}(\tau)$ is a count of how many times a trajectory transitioned from microstate i to j. A transition matrix $\mathbf{T}(\tau)$ can then be formed by dividing each count $\mathbf{C}_{ij}$ by the total number of observations of state j. More sophisticated approaches are now in use, the most common being to produce a model from

the largest connected set of states, using maximum likelihood estimation and Bayesian analysis[55, 62, 63]. In a Markov model, there may be thousands of microstates, and so one often chooses to coarse-grain the model at this point into some macrostates to make it more human-readable. Coarse-graining a model is not a trivial operation as combining microstates into macrostates may affect the quality of the model, and thus should primarily be used for visualisation properties. The most common technique for coarse graining a model is Perron-cluster cluster analysis (PCCA), a technique based on clustering the microstates according to spectral analysis of the eigenvectors of the transition matrix[54, 64].

There are many validation schemes for testing a Markov model. Most validation schemes are based on the Chapman-Kolmogorov equation[54, 57, 65]:

$$\mathbf{T}(n\tau) = \mathbf{T}(\tau)^n,$$

where n is an integer number of steps, and τ is the lag time. The basis of this equation is that taking n steps in a Markov model with lag time τ (the right-hand side of the equation), should be equivalent to taking one step with a lag time $n\tau$. The first test usually applied is to plot the implied timescales, which should be constant with increasing lag time τ above the Markov time - the time that exhibits the Markov property. The implied timescale, t_i, for an eigenvalue λ_i of the transition matrix is given by

$$t_i = -\frac{\tau}{\ln \lambda_i}.$$

If the implied timescales do not level off, this is an indication of either poor discretisation or poorly converged sampling. This method can also be used to identify an appropriate lag time for a model. Additional tests that are less subjective exist, and are detailed in Refs [54, 57].

Another general test that can be applied is bootstrapping, the process of rebuilding the model with subsets of the data. A converged model should be robust to the use of a random sample of the data[8].

These evaluations of a model are crucial, as, in the process of building a Markov model, there are many parameters and methods to tune. The derivation of a variational principle for Markov state models has enabled automated hyper-parameter optimisation procedures to begin to be developed[66], but this is not yet a fully automated procedure.

Open source tools for building Markov state models such as MSMBuilder and PyEMMA make the process of building and evaluating Markov models for the analysis of molecular dynamics trajectories accessible[67, 68]. By being able to produce high-level models

from large complex sets of molecular dynamics trajectories, they are a powerful method for the analysis of molecular simulation.

2.2 The Rare Event Problem

Propagation of the system in phase space through an approximate integration method means that each dynamics trajectory is a sample from the set of all possible trajectories, each of which is sensitive to tiny differences in initial conditions. This is the principle known as Lyapunov instability[24]. In running a dynamics simulation, it is assumed that the interactions between the atoms will give rise to the events that we hypothesise should occur, relying upon our choice of initial conditions and fluctuations in the underlying potential energy surface. The problem is that events such as protein rearrangements, enzyme-catalysed reactions and even simple reactions in the gas phase occur rarely[46, 69], and time-scales of trajectories are typically much shorter than the time required for these events to occur.

The reason for this is because many interesting events require the trajectory to transition over one or more energy barriers between states. A common representation of this for a chemical reaction is the free energy along some reaction coordinate which defines progress between states, as shown in Figure 2.2a.

The height of the barrier between states, $\Delta G^{\ddagger}$, determines the rate at which the reaction will happen. Under the assumptions of transition state theory (TST)[70], that there exists a hypersurface in phase space that divides reactants and products which has no 'recrossing' of products back to reactants before equilibration across this hypersurface, a relation between the rate of the reaction, k, and $\Delta G^{\ddagger}$ is given by the Eyring equation[71]:

$$k(T) = \frac{\Gamma(T) k_b T}{h} \exp\left(\frac{-\Delta G^{\ddagger}}{RT}\right),$$

where k_b is the Boltzmann constant, T is the temperature, h is Planck's constant, R is the molar gas constant, and $\Gamma(T)$ is the transmission coefficient. This equation shows that the rate of a reaction decreases exponentially as the height of the barrier increases. In classical TST, the transmission coefficient $\Gamma(T)$ is assumed to take the value unity, in which case $\ln(k(T))$ and $1/T$ have a linear relationship[72]. Table 2.1 provides some examples of how the reaction rate decreases as a function of increasing barrier height. The half-life, the expected time in which half of an ensemble would transition from reactant to product, is given by $\ln(2)/k$ and is shown in Table 2.1 for each barrier height. The key observation here is that in order for half of an ensemble of molecular dynamics

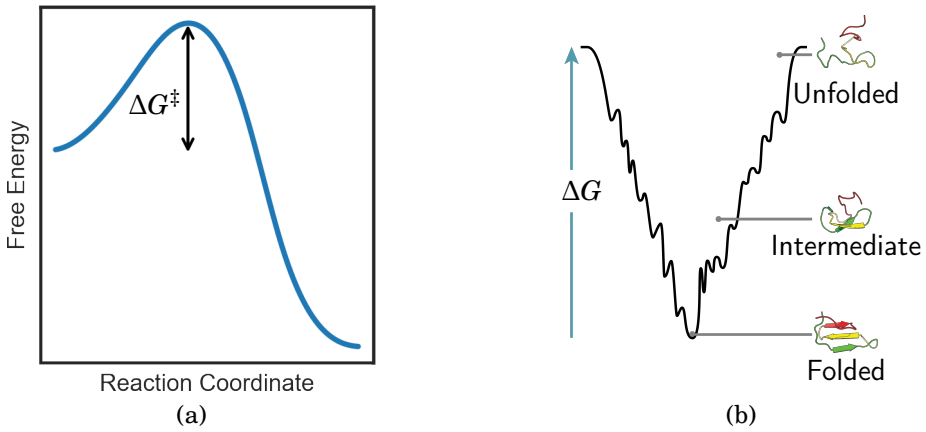


Figure 2.2: A) Illustration of a typical chemical reaction barrier with barrier height $\Delta G^{\ddagger}$, in which the reaction proceeds along some reaction coordinate. B) Schematic representation of the funnel-based protein folding landscape. Figure modified from figure by Thomas Splettstoesser licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

trajectories to overcome the energy barrier, the length of these trajectories increases exponentially with the barrier height. Large-scale simulations run on today's hardware access timescales on the order of microseconds to a few milliseconds[51, 53, 73], and thus may be able to sample transitions with barriers on the order of 10 kcal/mol. Many systems of interest are known to exhibit slower rates. For example, protein-ligand binding timescales may be up to milliseconds, and unbinding on the order of seconds[74], while protein folding events range from microseconds to seconds, generally correlated with the size of the protein[75].

The protein folding energy landscape does not have the simple shape of a chemical reaction. The current consensus is that the landscape is a rugged funnel[76], generally trending down in free energy from the unfolded state towards the folded state, as shown in Figure2.2b. This funnel shape explains the tendency for proteins to reliably fold into their native state, as the number of accessible configurations decreases as folding takes place. However, the ruggedness of the landscape means that many successive energy barriers between metastable states must be overcome en route.

Thus the crux of the rare event problem is that the phase space of a molecular system is divided into volumes of high-probability metastable regions (the wells in Figure 2.2), with low-probability transition regions between them (the peaks in Figure2.2). Molecular dynamics simulations, driven by the forces that derive from the gradient of the underlying potential energy surface, spend the majority of their time in these wells

$\Delta G^{\ddagger}$ (kcal / mol)	Reaction Rate (s^{-1})	Half-life
1	1.168e+12	593 fs
5	1.424e+09	486 ps
10	3.244e+05	2.13 μs
15	7.390e+01	9.37 ms
20	1.684e-02	4.11 s
25	3.835e-06	50.2 hours

Table 2.1: The relation between the barrier height, $\Delta G^{\ddagger}$, the reaction rate k, and the half life.

with very occasional transitions between metastable states. This means that new configurations in different metastable states can be hard to discover, and even if they are discovered, transitions are sampled rarely. In these situations, the ergodic hypothesis described by equation 2.7 is not satisfied, and so observables do not converge within accessible simulation timescales.

While continuous improvements in computational power mean trajectory lengths will continue to increase, the self-fulfilling prophecy of Moore's 'law' shows signs of slowing as physical limits of heat dissipation on silicon are reached. Today, builders of exascale machines are primarily concerned with maximising energy efficiency and fault tolerance[77]. Power and cooling constraints mean that clock speeds are not increasing, and instead performance is achieved by increased parallelism and a more complex memory hierarchy[78–81]. The development of general purpose graphics processing units (GPGPUs), and their relatively rapid adoption in numerous molecular dynamics packages[40, 82–85] is an indication of the scientific community's willingness to invest in diverse technologies to produce any cost reduction in simulations. Despite the highly optimised implementations with support for distributed parallel architectures achieved by some MD packages[84, 86], there is a limit to the parallelism that can be exploited in a molecular dynamics simulation, as there must be some communication and synchronisation of atom positions every time step. Furthermore, molecular dynamics trajectories are serial in time, as the results of the previous step must be computed in order to compute the next.

Some researchers have attempted to increase trajectory times by building specialised, dedicated machines for molecular simulation, such as the Anton machines[53, 73]. The second generation of these machines achieve millisecond long protein folding trajectories in two weeks; a speed-up of two orders of magnitude compared to traditional cluster

architectures.

While the achievements of these machines are impressive, the fact remains that observing a single folding event of a 10,000-atom simulation after two weeks of simulation does not produce converged, statistically significant sampling.

In contrast to these specialised machines, the other main approach in molecular dynamics has been the development of analysis methods for datasets comprised of many short trajectories, combining the developments of grid computing, cloud computing and Markov State modelling[8, 54, 66, 87]. These methods add an element of statistical rigour to the process of analysing molecular dynamics trajectories. However, sampling is still restricted to relatively small systems such as ligand binding events, loop motions, or the folding of short peptide chains[8, 88], as each short simulation is unlikely to traverse large energy barriers. These methods are discussed in greater detail in Chapter 7.

2.3 Accelerated Molecular Dynamics

To address the rare event problem, numerous accelerated MD methods have been developed that effectively enable one to sample structures that would otherwise only be possible on longer simulation timescales. The basic principle of the methods is to alter the sampling by manipulating some variable that affects what is sampled, *biasing* the simulation.

There are three broad categories of methods that take different approaches to achieve this. These are:

- Temperature based methods, which exploit the fact that raising the temperature of the simulation will make the resulting simulation overcome energy barriers more easily.
- Statistically biased methods, which exploit knowledge about the system to more optimally spawn new trajectories in poorly sampled regions.
- Potential energy-based methods, which apply a bias to the potential energy that governs the dynamics in order to encourage the system to overcome energy barriers.

Many methods have been developed that fall into one of these categories, as well as some that blend the ideas of multiple categories. Other factors that distinguish these

methods is the amount of information that needs to be known about the process of interest to use them, and the observables that can be computed with them. Some methods aim to capture all of the kinetic aspects of a system, while others aim to compute an important observable, such as the free energy of the event sampled. Some methods require very little knowledge about the system, while others assume that the free energy along a specific set of collective variables is all one wishes to compute.

An additional consideration is the unbiasing procedure of the method. Simply sampling lots of different states with a biased simulation may be qualitatively informative, but to calculate accurate observables it must be possible to calculate the underlying observables of the original, unbiased system robustly.

In this section, some of the representative methods will be reviewed, with a particular focus on free energy calculation.

2.3.1 Temperature Based Methods

An obvious and intuitive way to sample phase space more efficiently is to increase the temperature of the simulation. At higher temperatures, the system has more kinetic energy with which to overcome energy barriers, and so will sample phase space more rapidly. This is indeed a reasonable method in some cases, such as producing unfolded states in a protein folding simulation[89]. However, by exciting all the vibrations of the system, one may sample a lot of irrelevant states. The replica-exchange molecular dynamics (REMD) methods, such as parallel tempering, provide a way of estimating the equilibrium population of states by running simulations at lots of different temperatures and exchanging configurations between the simulations at these different configurations with a Monte Carlo step[90]. This enables low-temperature simulations to transition to those sampled at high temperature, and vice versa. This method has had some success in protein folding simulations and is effective for estimating the population of states, but kinetic properties are difficult to extract[91]. Furthermore, the method is computationally expensive[92], and the number of simulations and temperatures to run have to be chosen such that there is sufficient overlap between configurations at different temperatures, so that efficient mixing of the configurations in simulations offsets the cost of running many parallel simulations.

An additional problem with the method is that the probability of exchange decreases with increased dimensionality of the system, as each degree of freedom in the system reduces the probability of overlap in configurational space such that the Monte Carlo criterion for exchange is satisfied. A method to alleviate this in simulations with explicit

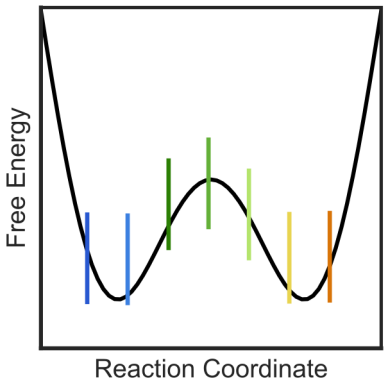


Figure 2.3: Schematic illustration of the phase space partition rare event methods such as BXD.

solvent has been developed, called Replica Exchange with Solute Tempering (REST), and its successor, REST2, in which the Hamiltonian of each replica is altered such that acceptance criteria for replica exchange does not depend on the number of water molecules [93, 94]. Another strategy is to combine the method with another biasing method, either along collective variables or the energy function itself. Such methods are described below in the context of metadynamics.

2.3.2 Ensemble Based Methods

Another approach is to statistically bias the simulation, through the use of an ensemble of simulations specifically selected to sample the less well-sampled regions of phase space.

Transition path sampling (TPS), and its derivatives such as transition interface sampling (TIS), are the progenitors of this category of methods[95, 96]. It is an importance sampling method, based on Transition State Theory, biasing trajectories based on Montecarlo acceptance criteria towards those that are reactive trajectories. Trajectories are spawned from the path, typically with random perturbations in velocities, and those that make progress along the reaction path, subject to a detailed balance criterion, are weighted more heavily for resampling. The method requires a good estimation of an initial path between reactants and products and requires many trajectories to converge. Transition Interface Sampling converges more quickly through the introduction of partitions along the path[96].

Descendants of the transition path sampling method use collective variables or a reaction coordinate to define a region of phase space captures the process of interest, and partition this space up into regions that are sampled independently, by producing trajectories that reach the boundaries between the partitions (see Figure 2.3). The acceleration comes from the fact that regions of phase space that would not be sampled regularly in unbiased molecular dynamics, such as transition states across energy barriers, can be repeatedly sampled by initialising trajectories in the region.

Forward flux sampling (FFS), milestoning, the weighted ensemble and boxed molecular dynamics (BXD) methods all fall under this category. The advantages of these methods are their abilities to calculate kinetic rates as well as free energies, and the fact that the underlying dynamics is still meaningful as no bias has been introduced in the potential energy surface.

In forward flux sampling[97], a reaction coordinate (or order parameter) is divided with a series of partitions $\rho_0, \rho_1, \ldots, \rho_n$, and an ensemble of trajectories are started from ρ_0. Any that reach ρ_1 have their configurations stored. These configurations are then chosen at random and new trajectories are spawned that either reach ρ_2 or return to ρ_1. From this, the probability of reaching ρ_2 can be calculated. The process is repeated until ρ_n is reached, at which point reaction rates can be calculated. Forward flux sampling has the advantage that, unlike TPS, it can be used for non-equilibrium processes, but many trajectories have to be used. The weighted ensemble method is very closely related to FFS[97, 98], but uses a hierarchical scheme to more efficiently coordinate the sampling of the reaction coordinate space[99]. It has been used to accelerate very long timescale events, including a ligand binding event on the order of minutes[100].

In the boxed molecular dynamics method[69], the reaction coordinate is again divided with a series of partitions, and trajectories are run in each partition. The key difference is that rather than reinitialising trajectories when they reach a partition boundary, the trajectories are continued by inverting the velocities of the atoms in the trajectory. The velocity inversion procedure conserves energy as well as linear and angular momentum, and so minimally perturbs the underlying trajectory. This means that dynamics can be constrained within poorly sampled regions until convergence is achieved, but introduces the requirement that trajectories must decorrelate between reflections against a boundary. The method is discussed in more detail (and improved upon) in Chapter 3. Like FFS, it can be used in non-equilibrium regimes [69, 101].

BXD shares several similarities with the milestoning method[102], which partitions a reaction coordinate with 'milestones', boundaries which represent an infinitesimal

slice of phase space. In the original formulation, trajectories had to be initialised at equilibrium on a milestone and were run until a subsequent milestone was reached. A more recent update of the procedure uses Voronoi tessellations to partition collective variable space[103], with trajectories locked between milestones using a velocity inversion procedure. The inversion procedure used differs from the BXD method in that it is merely an inversion of all the Cartesian velocities of the atoms, so it does not conserve linear and angular momentum.

Another importance sampling approach has been developed that leverages the analysis of Markov state models[8, 54]. By building a Markov state model after some short molecular dynamics trajectories, the regions of phase space that have been observed but not adequately sampled can be used to reinitialise sampling. Thus, the model adaptively expands across phase space, with each generation filling in the statistical gaps of the previous generation. The method has been used successfully to accelerate small protein folding simulations, ligand binding and protein-protein associations[8, 104].

The advantage of this approach is that one does not have to identify specific paths or collective variables before running simulations. However, since each trajectory is unbiased, the majority of simulation time will be spent exploring local conformations, and so it may not be as efficient as more focused methods.

2.3.3 Atom Potential Biasing Methods

The last of the three major categories of methods are those that modify the underlying potential encountered by atoms.

An obvious strategy is to steer a molecular dynamics trajectory from an appropriate and easy-to-define state towards the point of interest. This is precisely the method used in steered molecular dynamics[105] and targeted molecular dynamics[106, 107] by introducing additional forces to the potential. The former employs a time-dependent external force in a predetermined direction. A typical formulation is a harmonic potential that moves from an initial position $\vec{s}_0$ with constant velocity $\vec{v}$:

$$V_B(\vec{s},t) = K(\vec{s} - \vec{s}_0 + \vec{v}t)^2,$$

where $\vec{s}_0$ is the initial position, $\vec{s}$ is the current value of the collective variables, and K is a spring constant. Targeted molecular dynamics is analogous except it uses the root-mean-square distance (RMSD) between the current configuration and a target configuration to provide the direction in which to apply a harmonic potential[107].

Through the use of Jarynski's equation, which relates the amount of work done on a system to the change in free energy, it is possible to compute free energy profiles along a steered molecular dynamics simulation[108]. However, a high-spring constant K must be used, restricting the dynamics to follow the steered path very closely.

These methods saw several notable uses in exploring mechanical functions of proteins and binding simulations[109]. They have limited utility in more complex processes, however, as they effectively attempt to steer dynamics along a straight line in collective variable space. Such paths may be high in energy or difficult to sample.

The three other prominent methods used for biasing the potential energy surface are umbrella sampling, accelerated sampling, and metadynamics.

In the umbrella sampling method[47, 110], multiple bias potentials are added along a single collective variable s to overcome free energy barriers (depicted in Figure 2.4), with each potential typically taking the form of a harmonic potential:

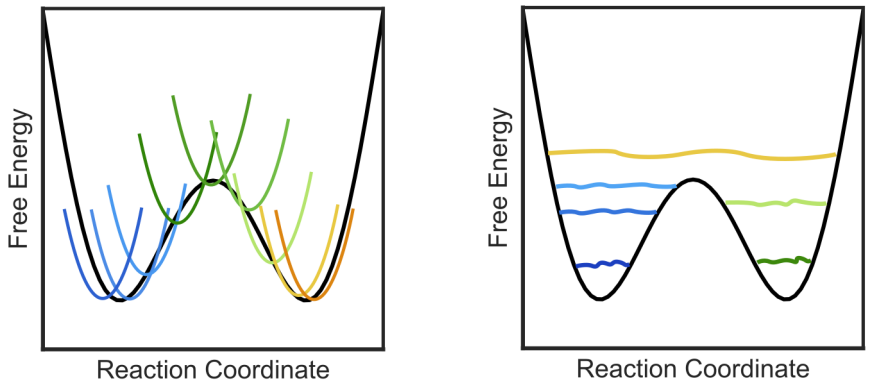
$$V_B(s) = K(s - s_i)^2,$$

where s is the collective variable value, s_i is the centre of the potential, and K is the spring constant.

These potentials confine the dynamics to a particular region, or 'window', of CV space. Multiple overlapping windows are used to enable sampling of the entire region, with dynamics run independently in each window. The statistics gathered from each of these simulations must be unbiased to determine the free energy of the original surface. The weighted histogram analysis method (WHAM) is widely used to achieve this[111]. Users of the method have to be careful of numerical error which can arise from poor sampling within a window, but the recent development of the dynamic weighted histogram analysis method (DHAM) may potentially alleviate some these issues[112]. The placement of windows and choice of the biasing potential requires a significant amount of trial and error, so an adaptive procedure was developed to automate this[113]. Umbrella sampling can be used simultaneously on multiple collective variables but scales poorly with increasing dimensions.

Metadynamics[114] is a related method that adaptively introduces bias potentials throughout the dynamics simulation to escape energy minima. The bias is built as a sum of Gaussian functions that are placed to fill domains in CV space that have been visited so far. The shape of the free energy profile $\Delta G(\vec{s})$ describes the goal of a metadynamics simulation. If one were able to construct a biasing potential $V_B(\vec{s})$ such that, up to an additive constant C,

$$\Delta G(\vec{s}) + V_B(\vec{s}) + C = 0,$$



(a) (b)

Figure 2.4: Schematic illustrations of A) the umbrella sampling method and B) the metadynamics method. A) A set of umbrella potentials across the underlying double well potential. B) A schematic illustration of a metadynamics trajectory depositing potentials until a diffusive surface is reached. Deposition starts on the left hand well and is coloured from blue through green and yellow as a function of simulation time.

then the resulting dynamics on potential would be that of a flat diffusive surface upon which a random walk would occur. On this surface, all of the relevant configurations and pathways would be sampled rapidly, and $\Delta G(\vec{s})$ is simply the negation of the bias potential (to an additive constant). In practice of course, $\Delta G(\vec{s})$ is not known, and so metadynamics attempts to construct biasing potentials that lead to as close to diffusive behaviour as possible.

Following the formulation presented in Ref [115], in a metadynamics simulation, the bias potential takes the form

$$V(\vec{s}(t),t)_B = \sum_{k\tau<t} W(k\tau)\exp\left(-\sum_{i=1}^{d}\frac{(s_i(t)-s_i(\vec{r}(k\tau)))^2}{2\sigma_i^2}\right),$$

where τ is the stride at which new Gaussian potentials are deposited, σ_i is the width of the Gaussian for the collective variable s_i, and $W(k\tau)$ is the height of the Gaussian.

In the infinite limit, the bias is approximately the negative of the free energy, $\Delta G(\vec{s})$. The original formulation of metadynamics had difficulties in determining when to stop a run. Since the method deposited Gaussians of a constant height, h, with $W(k\tau) = h$, it would over-fill the surface, sampling irrelevant regions of configurational space and oscillating around the true value of $\Delta G(\vec{s})$. These issues led the development of

well-tempered metadynamics[116], where the deposited Gaussian heights decrease over time as it explores the collective variable space:

$$W(k\tau) = W_0 \exp\left(-\frac{V_B(\vec{s}(\vec{r}(k\tau)), k\tau)}{k_B \Delta T}\right),$$

where W_0 is the initial Gaussian height and ΔT is a temperature.

Metadynamics is implemented in PLUMED[117], which allows it to be used in a wide variety of molecular dynamics packages. As a result, it has seen widespread adoption in a broad range of applications[47].

A limitation of metadynamics is that it must sample large regions of CV space in order to fill the free energy surface with bias potentials. This means it scales poorly with the dimensionality of the CV space, which limits its application to systems which can be described by a small number of collective variables. A number of new variants of metadynamics have been developed to that attempt to resolve the problem[92], including bias exchange metadynamics[118, 119], which combines metadynamics runs in many collective variables with replica exchange molecular dynamics, allowing fast exploration along each CV. Well tempered metadynamics using the potential energy as the collective variable, the so called Well Tempered Ensemble, has also been combined with parallel tempering to enable rapid sampling of configuration space[120].

Another similar method, accelerated sampling[121, 122], does not require the definition of collective variables at all. Instead, a boost potential is applied to the potential energy surface whenever the energy is less than some predetermined value E:

$$V(\vec{r})_B = \begin{cases} V(\vec{r}) & \text{if } V(\vec{r}) \geq E \\ V(\vec{r}) + \Delta V(\vec{r}) & \text{if } V(\vec{r}) < E. \end{cases}$$

While the exact form of $\Delta V(\vec{r})$ varies[122], a common choice[121] is

$$\Delta V(\vec{r}) = \frac{(E - V(\vec{r}))^2}{\alpha + (E - V(\vec{r}))},$$

where α is a tuning parameter. The advantage of this method is of course the fact that collective variables do not need to be defined, but, like parallel tempering, the parameters must be carefully tuned so as to explore relevant regions of conformational space. It is often combined with the use of restraining potentials so that only relevant parts of phase space are explored[123].

2.4 Discussion

There is a huge array of methods for overcoming the rare event problem, all with their advantages and shortcomings. The general trade-off one is making in these methods is the amount of information required versus the computational resources available[124]. Methods such as parallel tempering or accelerated molecular dynamics can sample many configurations but are computationally expensive or may sample irrelevant regions of phase space. Methods based on ensembles of trajectories can be used to calculate kinetic information or be used in non-equilibrium regimes, but are either slow or require the definition of an appropriate collective variable. Collective variable methods based on biasing the potential energy surface, such as metadynamics, are efficient at sampling and converging free energy surfaces, but, like all collective variable methods, requires careful selection of a small set of collective variables that adequately describe the process[46, 92]. The prevailing method to identify collective variables has been to exploit knowledge about the system: the user knows the process that they want to sample, and so can generate a set of candidate collective variables using 'chemical intuition'. This approach is not particularly systematic and does not leave room for the discovery of better collective variables than that constructed by the user.

In recent years, as the size of molecular systems being studied has increased there has been a trend to introduce more general dimensionality reduction techniques from the fields of machine learning such as principal component analysis (PCA), time-lagged independent component analysis (TICA), diffusion maps[125], and even neural-network based variational auto-encoders[54, 61, 126]. Furthermore, there has been some success in combining these methods with accelerated sampling methods[127]. One problem with these methods is that the interpretable, human-readable nature can be lost, and thus there have been attempts to select more intuitive reaction coordinates that best explain the underlying data[128]. Another strategy is to represent processes with a series of configurations, called paths, avoiding the need to identify the collective variables at work explicitly[129].

These approaches are appealing, as they provide a way to automate the process of finding collective variables. A drawback is that they require data to work with, either in the form of a molecular dynamics trajectory from which to extract variables[128, 130], or an initial path demonstrating the process that is to be sampled further[129]. This leads to something of a chicken-and-egg situation, as in order to find collective variables which allow efficient sampling of an event, the event needs to either have already been

observed or at least be described.

In the following chapters, these issues are discussed in more detail from the context of different methods and applications. The boxed molecular dynamics method is extended and generalised, making it more competitive with existing methods and bringing its advantages with it. Attention is then turned to the problem of finding initial datasets from which to extract initial conditions, collective variables, and paths. Here, advances in virtual reality technology are exploited to develop an interactive molecular dynamics framework that allows users to perform accelerated initial sampling of molecular processes.

CHAPTER 3

EXTENDING BOXED MOLECULAR DYNAMICS

3.1 Boxed Molecular Dynamics

As discussed in Chapter 2, Boxed Molecular Dynamics[69, 101] (BXD) is an enhanced sampling algorithm that uses a collective variable to accelerate rare-event dynamics. In this chapter, the original algorithm is reviewed in detail, and novel algorithmic developments and applications are described and evaluated that culminated in publication as Ref [131].

In its initial formulation, BXD accelerates dynamics by introducing a series of constraints along a one-dimensional collective variable (reaction coordinate), which provide a set of 'boxes' within which to lock the trajectory, as illustrated in Figure 3.1. The region defined by the reaction coordinate ρ is split into m boxes between points B_R and B_P by the introduction of $m+1$ user-defined constraints. The trajectory is kept in each box by an elastic collision procedure with the boundaries. Whenever the next time step in the dynamics would result in the trajectory crossing the boundary, the trajectory is reset to the previous step, and a velocity inversion procedure is applied to those atoms that contribute to the definition of the reaction coordinate. Constraining the dynamics in this manner means that configurations that may be unfavourable can be sampled more thoroughly, and using multiple boxes allows one to coax dynamics along pathways of interest. The fundamental speed-up comes from the fact that it is easier to converge statistics within each box separately than the total configuration space.

After sufficient sampling within each box, rate coefficients and free energies may

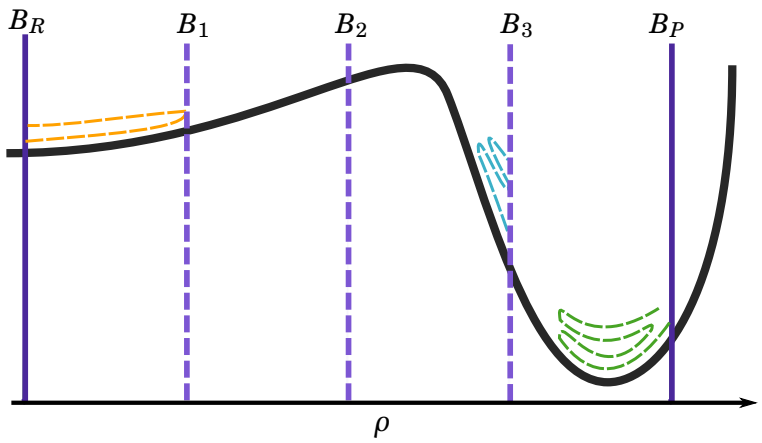


Figure 3.1: Illustration of the original BXD algorithm. A reaction coordinate ρ is partitioned between regions B_R and B_P with a series of hard boundaries, B_1, B_2, and B_3, in which trajectories are locked through a velocity inversion procedure.

be calculated as follows. For a given box i bounded by ρ_i and ρ_{i-1}, the rate coefficient for transfer from box i to $i-1$ is determined by the inverse of the mean first passage time (MFPT), $\langle \tau_{i,i-1} \rangle$. This can be computed by keeping track of the number of times the trajectory is inverted at each boundary, and the lifetime of the trajectory within the box, which gives the rate coefficient for transfer from box i to box $i-1$, $k_{i,i-1}$ via:

$$k_{i,i-1} = \langle \tau_{i,i-1} \rangle^{-1} = \frac{h_{i,i-1}}{t_i}. \tag{3.1}$$

Equilibrium constants between box i and box $i-1$ may then be obtained from equilibrium statistical mechanics[69] as

$$K_{i-1,i} = \frac{k_{i-1,i}}{k_{i,i-1}} = \exp\left(\frac{-\Delta G_{i-1,i}}{k_B T}\right), \tag{3.2}$$

where $\Delta G_{i-1,i}$ is the free energy difference between box i and box $i-1$. With respect to an arbitrary zero, the free energy of each box ΔG_i may then be determined along with p_i, the probability of the trajectory residing in box i:

$$p_i = \frac{1}{\sum_i \exp(-\Delta G_i / k_B T)} \exp(-\Delta G_i / k_B T). \tag{3.3}$$

With the probability calculated for each box, it is then possible to determine $p(\rho)$ to arbitrary resolution by renormalising statistics within each box using histogram binning. By estimating $p_i(\rho)$, the probability of a particular value of ρ within the box, by

constructing a histogram from observations over the trajectory, the probability of residing anywhere along the reaction coordinate is then

$$p(\rho) = p_i(\rho) \times p_i. \tag{3.4}$$

With this in hand, the free energy along the reaction coordinate to arbitrary resolution can be obtained via

$$\Delta G(\rho) = -k_B T \ln(p(\rho)). \tag{3.5}$$

Since the box-to-box rate coefficients are the only values that need to be computed, the length of time the trajectory needs to spend in each box is only determined by how long it takes for these rate coefficients to converge.

The dynamics within each box can be sampled independently, which lends itself well to parallelisation on modern cluster architectures, as shown in Figure 3.1, in which a trajectory is shown running independently in each box. Alternatively, it is easy to formulate a trajectory such that after a given number of inversion events with a boundary the trajectory is allowed to proceed to the next box.

BXD has some advantages over some of the other enhanced sampling methods described above. Foremost, it is a simple and intuitive method, making it easy to implement in dynamics packages and easy to interpret. As demonstrated above, and unlike some of the other methods described previously, it is possible to compute both thermodynamic and kinetic information in the form of rate coefficients and free energy respectively from a single run. BXD does not perturb the underlying potential energy surface, and the velocity inversion procedure may be formulated so that it conserves energy as well as linear and angular momentum, which means that, following decorrelation of velocities after inversion against a boundary, the dynamics within a box are meaningful. Maintaining physically meaningful dynamics allows the BXD method to be used within a microcanonical (NVE) ensemble, where the number of particles (N), volume (V), and energy (E) are fixed[132, 133].

The original BXD implementation had several aspects that limit its utility. Firstly, the procedure requires an appropriate selection of boundaries along the reaction coordinate. The user places these boundaries through a combination of experimentation and knowledge of the system. They then perform trial molecular dynamics trajectories, iteratively adjusting the positions of the boxes until simulations reach the boundaries of all boxes in a reasonable amount of time. To date, users of BXD have exploited the fact that the results it generates are mostly insensitive to the position of the boundaries. So long as the transit time from one box boundary to another is larger than the

system's decorrelation timescale, which is the time it takes for the dynamics within a box to be independent of the previous boundary hit, the statistics generated are largely invariant[69, 101]. However, this does not mean that the time required to converge passage times is independent of the boundary placements, as some regions of the reaction coordinate space require more acceleration than others.

Additionally, the original formulation of BXD allows for the use of only one collective variable, referred to as a reaction coordinate. The requirement for the rare event of interest to be represented by a one-dimensional reaction coordinate restricts BXD to simple events. In what follows, novel algorithmic developments of the BXD method are described and demonstrated through application in sampling a chemical reaction in a liquid.

3.2 Adaptive Boxed Molecular Dynamics in 1D

Before presenting the full generalisation of BXD to multi-dimensional collective variable space, let us begin with an automated procedure for placing boundaries along a single reaction coordinate. Automating the placement of boundaries is a natural extension of the original BXD algorithm, stemming from the simple desire to avoid the labour in placing boundaries by hand. Moreover, it provides the context in which generalisations can be made.

For a one-dimensional surface represented by ρ, the reaction coordinate, the width of the BXD boundaries should be inversely proportional to the gradient of the surface, as illustrated for a hypothetical surface in Figure 3.1, where boundaries have been placed equally spaced along the reaction coordinate.

If a region has a large gradient, the box needs to be smaller so that it is constrained to sample the steep incline. An unbiased trajectory is inclined to travel downhill rather than uphill and so if too large a box is placed across the region the higher part of the slope will rarely be sampled. The blue and green trajectories of Figure 3.1 bounded by B_2 and B_3, and B_3 and B_4 respectively, illustrate this problem. Hence to accelerate sampling on the entire slope of the hill we need to place several small boxes.

Conversely, if the region is relatively flat, then the box should be larger so the trajectory can sample the whole region rapidly. As shown by the trajectory in the first box in Figure 3.1, created by boundaries B_R and B_1, a box that is too small unnecessarily constrains the dynamics within the region, when it could be sampling other areas. In extreme cases, small boxes can result in dynamics that does not decorrelate between

successive inversions, biasing the statistics.

In what follows, an automated algorithm is described which attempts to automatically determine optimal box locations through on-the-fly statistical analysis during a trajectory, published in Ref [131].

Instead of providing a full list of hand-picked box boundaries, the user provides the following input data which they believe adequately represents an educated guess pertinent to the rare event they wish to accelerate: (1) a reaction coordinate definition $\rho \in \mathbb{R}$ and (2) a minimum and maximum bound on that coordinate. Let $\Gamma \in \mathbb{R}$ be the region of phase space defined by the two boundaries B_R and B_P. The approach in adaptive BXD is to make two passes over the space, once ascending from B_R to B_P and then descending back down from B_P to B_R placing additional bounds into an ordered set of boundaries, $\{B_R, B_1, \ldots, B_m, B_P\}$, as necessary so as to proceed in a timely fashion. A pass in both directions is required so that any slopes in the energy landscape are sampled in both directions - a downhill slope in one direction will require additional bounds from the other direction. Upon completion of the two passes, Γ is partitioned into a set of boxes with bounds for subsequent use in the BXD algorithm. The overall process is illustrated in Figure 3.2 for a typical hypothetical reaction profile described by reaction coordinate ρ.

The procedure for identifying intermediate bounds is illustrated in Figure 3.3. Without loss of generality, we assume that sampling is heading for B_P from B_R. At the start of the adaptive trajectory we define $B_i \leftarrow B_R$ and $B_{End} \leftarrow B_P$, indicating that B_i is our current best boundary on route to the target destination, B_{End}. The trajectory samples the region of Γ between B_i and B_{End}, Γ_s, for n steps, constrained to stay within the region by the velocity inversion procedure of BXD. After n steps, there are two possible outcomes: (1) there are no velocity inversions against B_{End}, indicating that Γ_s could be more efficiently sampled with an additional BXD boundary, or (2) velocity inversions against B_{End} were observed, indicating that there is no need for additional boundaries in Γ_s as the sampling of the space is occurring efficiently. In the first case this additional boundary B_{new} is placed based on the values of ρ sampled (the procedure for which is detailed below), and $B_i \leftarrow B_{new}$ to sample the region enclosed between B_{new} and B_{End}. In the second case, if $B_{End} = B_P$ then the target has been reached, and so the direction of travel is reversed (see Figure 3.2b), with $B_i \leftarrow B_P$ and $B_{End} \leftarrow B_{i-1}$. Sampling occurs again, but this time there are additional boundaries $B_R, B_1, \ldots, B_m, B_P$ along the route. During the second pass, if inversions against B_{End} occur, then either $B_{End} = B_R$, or $B_{End} = B_{i-1}$, the latter being some intermediate boundary placed during the first pass.

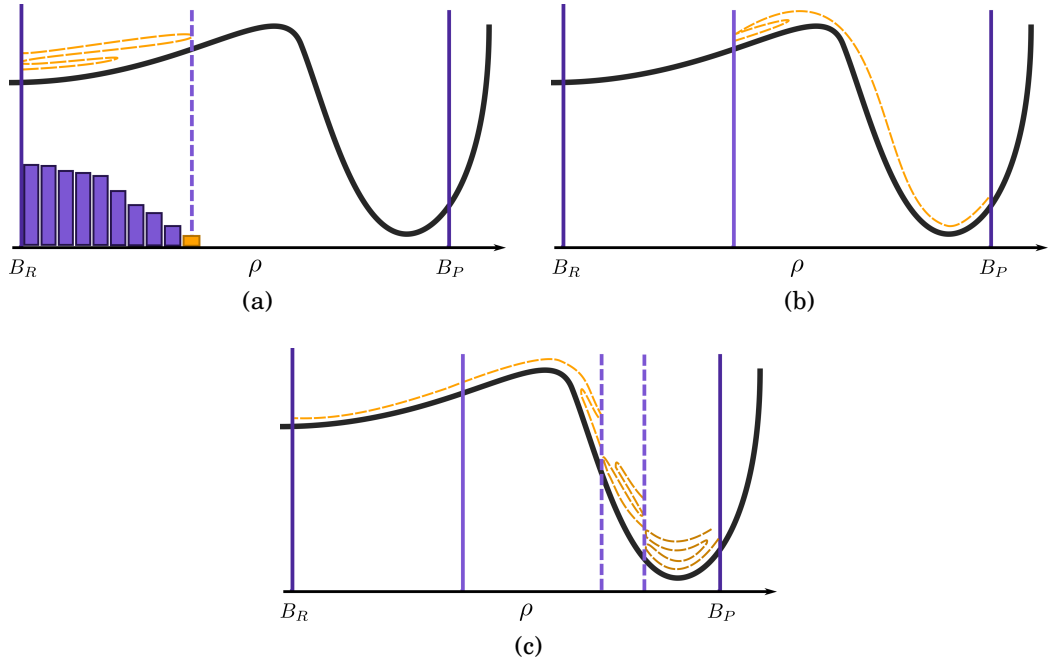


Figure 3.2: Illustration of the adaptive BXD boundary generation algorithm. A) Placement of the first boundary based on a histogram of sampled values. B) Continued sampling from the placement of the first boundary resulting in a transition and inversion against B_P, upon which sampling direction is reversed. C) Placement of additional boundaries to accelerate the reverse reaction. The boundary generation algorithm is complete when the trajectory returns to B_R.

In the first case, we have sampled the whole space and are done. In the second case the region between B_i and B_{i-1} is well sampled, so no additional boundary is placed, $B_{End} \leftarrow B_{i-2}$ and $B_i \leftarrow B_{i-1}$ (see Figure 3.2c).

When a new boundary is deemed to be required, it is placed by constructing a histogram of the n previously sampled values of ρ between B_i and B_{End} (see Figure 3.2a). The value of ρ maximally far from B_i with probability of being sampled at least $\epsilon \in (0,1)$, according to the histogram, is chosen as the position of the new boundary. The parameter ϵ is used to adjust the distance at which a boundary is placed. If the value of ϵ is too small, the boundary may be placed at some extreme value of ρ representing a rare event, resulting in poor acceleration. If ϵ is too large, the boundary may be placed too close to B_i, resulting in ballistic trajectories within the box in which dynamics do not decorrelate. So far, a value between 0.01 and 0.1 has been found to be suitable, and the results are not sensitive to the specific value.

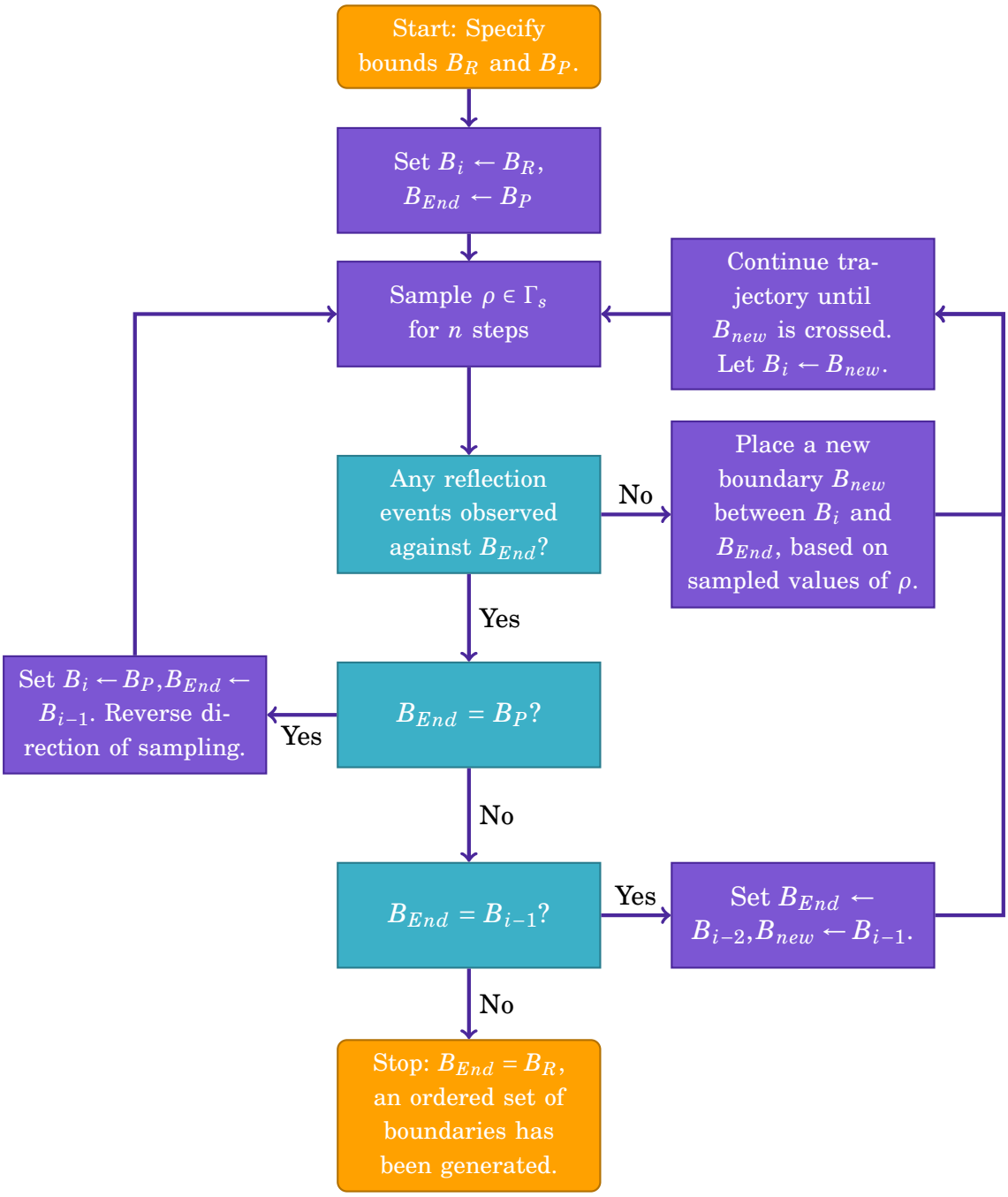


Figure 3.3: Flowchart illustrating the adaptive BXD boundary generation procedure.

3.3 Extending Boxed Molecular Dynamics to Multidimensional Collective Variable Space

The automatic boundary generation procedure described above makes it easier to run an accelerated BXD calculation for systems that can be described with a one-dimensional collective variable. However, in practice, many systems are not easily represented by a single collective variable, and so it is desirable to accelerate dynamics in multidimensional collective variable space.

BXD could be extended to support multidimensional CVs in a number of ways. An obvious manner, as suggested in Ref [69], is to partition the space with BXD boundaries in each dimension, in a similar manner to that taken in multidimensional umbrella sampling[134]. An illustration of this strategy is depicted in Figure 3.4a. While the simplest extension, this approach has many problems. Firstly, the number of partitions scales with the dimensionality of the CV space, making it inefficient for high-dimensional systems. This scaling property is also wasteful, as large regions of the CV space are likely to be irrelevant due to their high energy. Additionally, partitioning the system in a grid will ignore the gradient of the underlying potential, resulting in the inefficiencies described in vanilla BXD.

A more attractive option is to accelerate along pathways through the collective variable space, partitioning the space with BXD boundaries orthogonal to the direction of motion. In this approach, shown in Figure 3.4b, the number of partitions scales with the length of the path, and only the regions of CV space relevant to a given application will be sampled. For a system of N atoms described by a set of M collective variables represented as a vector $\vec{s}(t) = [s_1(t), s_2(t), \ldots, s_M(t)]$, the collective variable space is partitioned into a set of $(M-1)$-dimensional boundaries. A one-dimensional space (as in vanilla BXD) is partitioned by a series of points along the reaction coordinate; a two-dimensional CV space is partitioned by a series of lines, a three-dimensional CV space is partitioned by a series of planes, and so on to the general case of hyperplanes. In this form, a BXD boundary B_j is defined as a hyperplane in Hessian normal form with unit norm $\vec{n} \in \mathbb{R}^M$ and positioning constant D_j:

$$B_j \equiv \left(\sum_{i=1}^{M} n_i s_i \right) + D_j = 0. \tag{3.6}$$

This approach is a natural extension of the original BXD algorithm: passage times between a linear sequence of boxes is all that one needs to sample, and the calculation of

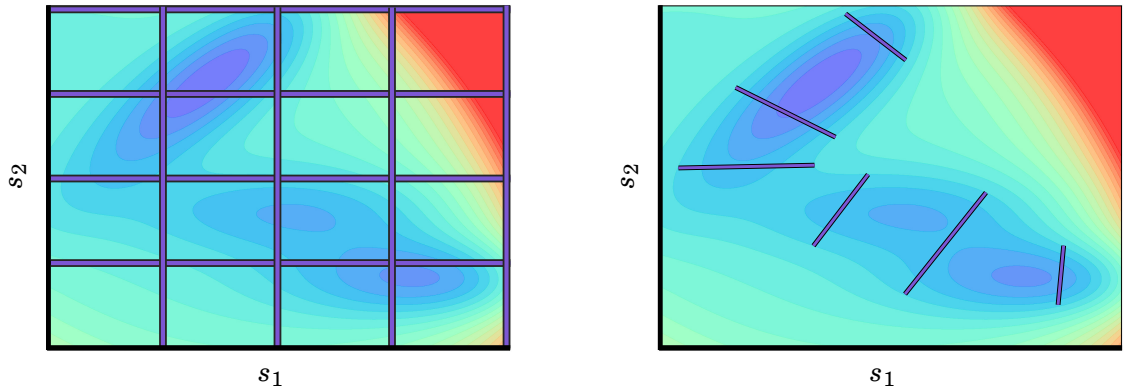


Figure 3.4: Strategies for extending BXD to multidimensional collective variable space. Panel A illustrates the grid-based approach in which the appropriate CV region is divided evenly, while panel B shows the path-based approach in which boundaries are placed orthogonally to the dynamical pathway.

box-to-box rates remains unchanged. The method requires a generalisation of the velocity inversion procedure. Additionally, the use of hyperplanes as boundaries compounds the problem of their placement, as one now needs to choose the location and orientation of the hyperplane in a higher dimensional manifold. The following sections address these challenges.

3.3.1 General Velocity Reflection Procedure in Multidimensional Collective Variable Space

Given a set of BXD boundaries as defined above, generalising the BXD algorithm requires a reformulation of the velocity inversion procedure. The original algorithm devised a method for inverting velocities along the reaction coordinate, minimally perturbing the system to ensure that a boundary was not crossed[69]. For several reaction coordinates, it had been formulated to conserve energy, as well as linear and angular momentum. However, the procedure had to be derived and implemented separately for each reaction coordinate, and there is no clear way to generalise to multidimensional collective variables. One approach is to simply invert the Cartesian velocities of all the atoms involved in the collective variable definition. This is the method used in milestoning with Voronoi tesselations[103], which shares similarities to the multidimensional generalisation of BXD. This approach does not minimally perturb the dynamics, how-

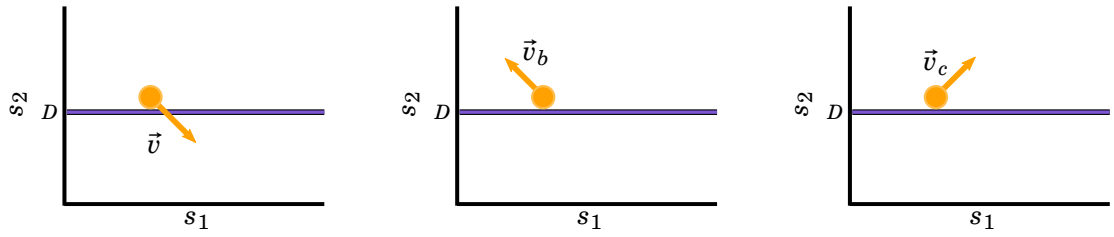


Figure 3.5: Diagram illustrating the desired properties of a velocity inversion in multidimensional collective variable space. Panel (a) shows a particle about to cross the boundary B, (b) shows the result of a simple Cartesian inversion of velocities and (c) shows a reflection against the boundary.

ever, as can be seen in the following toy model.

Consider a system represented by two collective variables s_1 and s_2, and suppose we constructed a boundary $B = s_2 - D$, i.e, a boundary that is only dependent on the value of s_2. As shown in Figure 3.5b, simply inverting the Cartesian velocities results in the propagation along s_1 to be inverted as well as the desired inversion of the component contributing to crossing the boundary. Contrast this with a reflection against the boundary, as showing in Figure 3.5c, where the propagation of s_1 is unperturbed. By only perturbing the components of the velocity that would result in crossing the boundary, we minimally interfere with the dynamics and allow full exploration along s_1.

The generalisation of the velocity inversion procedure to a velocity reflection procedure in M-dimensional collective variable space was developed by recasting the problem to one of using impulses to enforce constraints, using the mathematical framework of rigid body dynamics under constraints[135–138]. Enforcing constraints on the atomic motion in molecular dynamics simulations is a routine operation available in many molecular dynamics packages, such as the restriction of interatomic distances or angles through algorithms such as SHAKE[139], RATTLE[140], Settle[141] or their derivatives[142]. Such algorithms represent the constraint in the following form:

$$f(\vec{r}(t)) = 0, \tag{3.7}$$

where the f is some function of atomic coordinates. A constraint of this form is a holonomic constraint[135], and is continuously enforced. The constraints that BXD enforces on a system can be represented in a similar form. Let B_j be a BXD boundary, repre-

sented as in Equation 3.6. The function

$$\phi(\vec{r}(t)) = \vec{s}(t) \cdot \vec{n}_j + D_j \tag{3.8}$$

provides a measure of how far the trajectory is from the boundary at time t, with changes in sign representing a crossing of the boundary. To constrain the dynamics of the system such that the trajectory remains on a particular side of B_j, we require:

$$\phi(\vec{r}(t)) \geq 0. \tag{3.9}$$

The introduction of the inequality in the definition of the constraint, compared to that given in Equation 3.7, results in what is referred to as a unilateral constraint. Unlike a holonomic constraint, a unilateral constraint is only enforced at time steps where the constraint is unsatisfied. A simple example from rigid body dynamics is that of a ball bouncing on a surface under the influence of gravity: the constraint that the ball should not intersect through the surface only needs to be applied at the moment of impact.

Similarly, the BXD boundary needs only to be enforced at time steps where the trajectory would cross the boundary if unperturbed and is implemented as follows. Suppose at time t that $\phi(\vec{r}(t)) \geq 0$, and in the next time step $t+\delta t$ the trajectory crosses the boundary, resulting in $\phi(\vec{r}(t+\delta t)) < 0$. Following the original BXD procedure, we seek to revert the atomic positions to the previous step, $\vec{r}(t)$, and invert the velocities to give new velocities $\vec{v}'(t)$, which result in the constraint being satisfied at time $t+\delta t$. The velocities are inverted as follows. By the chain rule, the time derivative of the constraint function $\phi(\vec{r}(t))$ is given by:

$$\frac{\mathrm{d}\phi(\vec{r}(t))}{\mathrm{d}t} = \frac{\mathrm{d}\phi(\vec{r}(t))}{\mathrm{d}\vec{r}} \cdot \frac{\mathrm{d}\vec{r}}{\mathrm{d}t} = \nabla\phi \cdot \vec{v}(t). \tag{3.10}$$

In a general system of rigid bodies, there are many constraints which need to be resolved simultaneously, in which case $\nabla\phi$ is a matrix of K rows by $3N$ columns, where K is the number of constraints. In the path-based methodology described here only one BXD boundary can be crossed at a time, meaning we can restrict ourselves to the case of a single constraint. The gradient $\nabla\phi$ in Equation 3.10 thus represents a row vector, which enables us to derive an analytic solution. However, it should be noted that the method can also be used with any number of independent holonomic and non-holonomic constraints (such as that shown in Figure 3.4a), as recently demonstrated in Ref [124].

To ensure that the constraint will be satisfied at time $t+\delta t$, the inverted velocities must satisfy the following:

$$\nabla\phi \cdot \vec{v}'(t) + b = 0. \tag{3.11}$$

The term b is typically used to introduce friction or elasticity into a rigid body system. In order to conserve energy, we seek a fully elastic reflection of the velocities normal to B_j, so we set $b = \nabla\phi \cdot \vec{v}(t)$, resulting in the following equation that must be satisfied:

$$\nabla\phi \cdot \vec{v}'(t) + \nabla\phi \cdot \vec{v}(t) = 0 \tag{3.12}$$

The equation of motion for dynamics[135] under a single constraint may be written as:

$$\mathbf{M}\vec{a} = \vec{f} + \vec{g}, \tag{3.13}$$

where $\mathbf{M} \in \mathbb{R}^{3N\times 3N}$ is a diagonal matrix of atomic masses, $\vec{a} \in \mathbb{R}^{3N}$ is the vector of accelerations, $\vec{f}$ is the force vector from the MD gradient calculation, and $\vec{g}$ are the forces due to the constraint, given by

$$\vec{g} = -\lambda\nabla\phi^T, \tag{3.14}$$

where λ is a time-dependent Lagrangian multiplier, and ϕ^T represents the transpose of the row vector ϕ. To act upon velocities rather than forces, the constraint is enforced using an impulse[137, 138] as follows:

$$\vec{v}'(t) = \vec{v}(t) + \lambda\mathbf{M}^{-1}\nabla\phi^T \tag{3.15}$$

By substituting Equation 3.15 into Equation 3.12 and rearranging for λ we have

$$\lambda = \frac{-2\nabla\phi \cdot \vec{v}(t)}{\nabla\phi\mathbf{M}^{-1}\nabla\phi^T}. \tag{3.16}$$

This value for λ can then be used to give $\vec{v}'(t)$, resulting in velocities that satisfy the constraint and reflect away from the boundary B_j. The Lagrangian multiplier and subsequent impulse need only to be computed and applied to time steps which if unaltered would result in crossing the BXD boundary, making it highly compatible with existing BXD implementations.

A demonstration of the inversion procedure for a toy system of 3 particles, A, B and C with a bond between atoms A and B, and a bond between atoms B and C, is shown in Figure 3.6A. The bonds are implemented as spring potentials, and the system is integrated under velocity verlet under NVE conditions (program available in Ref: [143]). Collective variables representing the A-B and B-C distances, s_1 and s_2 respectively, are constrained with a BXD boundary with unit norm $\vec{n} = (-0.37, 0.93)$ at distance $D = 0.5$.

The choice of the boundary location is arbitrary but includes contributions from both collective variables so that the inversion procedure can be tested thoroughly. Furthermore, this toy system is representative of abstraction reactions which are a target for acceleration (as explored further below). A derivation of the inversion procedure for a system such as this is given in Appendix A.

Figure 3.6 shows how a simulation under the new inversion procedure proceeds. The dynamics in collective variable space are shown in Figure 3.6B, where reflections against the boundary (represented as a dark grey line) are observed, exhibiting the desired reflective behaviour in all cases. Figure 3.6C, shows the value of $\phi(\vec{r})$ over the course of the trajectory. Whenever $\phi(\vec{r})$ reaches zero, the simulation has crossed the boundary, and the inversion procedure is applied. Figure 3.6D shows the energy of the system, and demonstrates that the new velocity inversion procedure conserves kinetic energy. While not shown, the procedure also conserves linear momentum.

The toy system demonstrates the effectiveness of the new inversion procedure for multidimensional collective variables. Furthermore, this inversion procedure is much more general than the original procedure, which had to be derived and implemented for each new reaction coordinate. The definition of BXD boundaries as hyperplanes with unit norm $\vec{n} = (n_1, n_2, \ldots, n_M)$ means that the derivatives of ϕ in Equation 3.10 may be computed as a linear combination of the derivatives of the individual components of $\vec{s}$:

$$\frac{\mathrm{d}\phi}{\mathrm{d}\vec{r}} = n_1 \frac{\mathrm{d}s_1}{\mathrm{d}\vec{r}} + n_2 \frac{\mathrm{d}s_2}{\mathrm{d}\vec{r}} + \ldots + n_M \frac{\mathrm{d}s_M}{\mathrm{d}\vec{r}}. \tag{3.17}$$

This feature is hugely beneficial for BXD's practical implementation. Requiring only the gradients of each collective variable means that it will now be much more straightforward to use with existing collective variable based methods and implementations, such as the PLUMED[117] package used for umbrella sampling and metadynamics.

3.3.2 Adaptive Boxed Molecular Dynamics in Multidimensional CV Space

With a definition of multidimensional BXD boundaries and a method for constraining dynamics within the regions defined by them, all that remains is to generalise the algorithm for placing BXD boundaries. The automated algorithm for generating BXD boundaries presented in section 3.2 generalises to multidimensional CV space straightforwardly. As in the one-dimensional case, a start and end point in the collective variable space is required, and then boundaries are placed in two passes over the system.

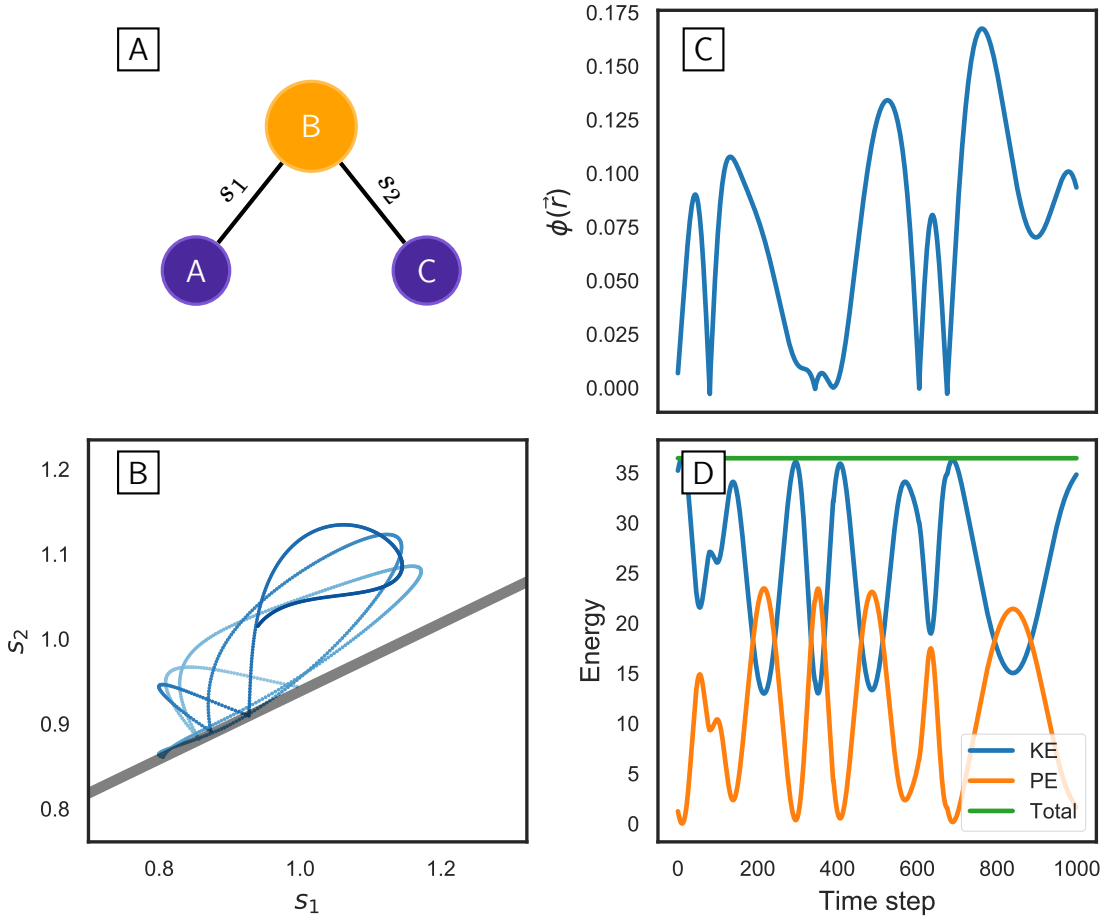


Figure 3.6: Demonstration of the generalised velocity inversion procedure for a toy system of three atoms with two harmonic bonds. A) A schematic of the A-B-C system with collective variables labelled. B) the trajectory projected onto the collective variables s_1 and s_2, coloured from dark to light blue over the course of the trajectory, with reflections against a BXD boundary (in grey) shown. C) Plot of the value of $\phi(\vec{r})$ over the course of the trajectory. D) The kinetic, potential and total energy of the system over the course of the trajectory.

The only significant change to consider is in how to place a new boundary. In the one-dimensional case, the method simply aimed to maximise (or equivalently, minimise) the value of ρ sampled so far, using a histogramming procedure. In the multidimensional regime, the value of $\vec{s}$ cannot simply be maximised or minimised, as individual components may be required to increase or decrease depending on the underlying path. How can we identify the correct direction in which to accelerate the dynamics, given what has been sampled so far? A greedy algorithm that extends naturally from the 1D case is proposed, which seeks to maximise the distance from the previous bound. Let $\mathbf{S} \in \mathbb{R}^{n \times M}$ be the set of sampled values of $\vec{s}$, as illustrated by the blue dots in Figure 3.7, and let $\vec{R} \in \mathbb{R}^n$ be the set of perpendicular distances from the boundary B_i to the sampled values in $\mathbf{S}$.

1. A normalised histogram of $\vec{R}$ is computed, giving an estimation of $p(r)$, the probability density function of the distance from B_i reached between successive velocity inversions (see Figure 3.7B).

2. The histogram bin that is maximally distant from B_i, subject to it having a cumulative probability of at least $(1-\epsilon)$ is identified as b_{max} and the average value of $\vec{s}$ in this region is computed to give $\vec{s}_{max}$.

3. Similarly, the average value of $\vec{s}$ in the bin closest to the bound B_i is computed to give $\vec{s}_{min}$. The normalised vector $\vec{n}^{new}$ from $\vec{s}_{min}$ to $\vec{s}_{max}$ is computed, providing the orientation of the new bound.

4. The new bound B_{new} is then defined with norm $\vec{n}^{'}$ with s_{max} lying upon it (Figure 3.7C):
$$B_{new} \equiv \left(\sum_{i=1}^{M} n_i' s_i \right) + D = 0, \text{ where } D = -\vec{n}' \cdot \vec{s}_{max}.$$

As illustrated in Figure 3.7, which is from a real application of a chemical reaction in a liquid (see Chapter 3.4), this procedure for generating a new BXD boundary allows the dynamics to follow the shape of the pathway of the underlying free energy surface. For the algorithm to be used in high-dimensional collective variable space, this is crucial, as it allows for the user to not know exactly the shape of the surface a priori - the only requirement is a start point and end point, providing a sense of direction.

With the adaptive algorithm and velocity inversion procedure generalised to the multidimensional collective variable case, BXD can now be used in a wide range of applications.

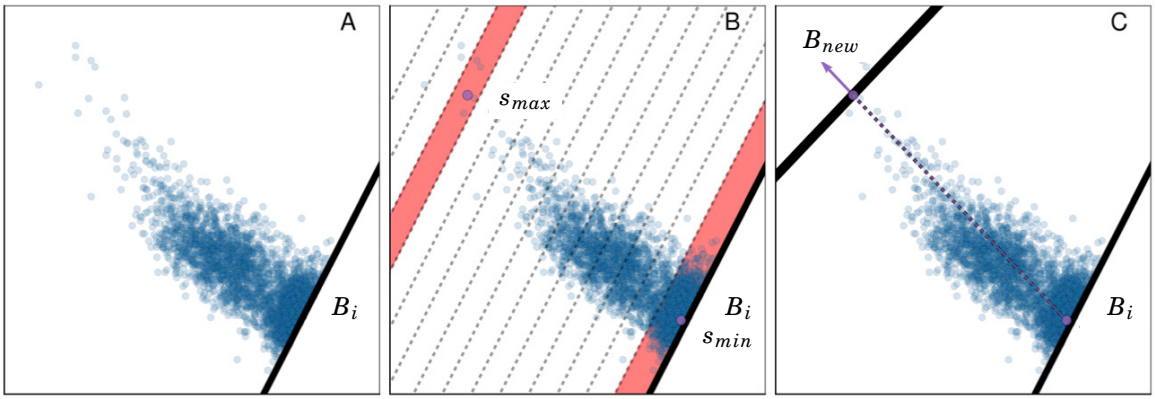


Figure 3.7: Illustration of the procedure used to generate a new intermediate BXD boundary in multidimensional BXD: A) Sampled values of $\vec{s}$, with boundary B_i used to constrain dynamics. B) Histogram binning of sampled values of $\vec{s}$. The bins used to identify s_{max} and s_{min} are highlighted in red. C) Placement of new bound B_{new}, using the vector from s_{min} to s_{max}.

3.4 Accelerated Sampling of Chemical Reactions in Liquids

In this section, the extensions to BXD described in the previous sections are evaluated through application to the reaction F + CD_3CN → DF + CD_2CN in both the gas and solution phases. The input files and resulting datasets are available in Ref [144]. This system is an appropriate test of the method, as it has been the subject of both ultrafast transient IR spectroscopy experiments and MD simulations[6], providing experimental values and previous theoretical values against which to compare. The reaction is the abstraction of deuterium by a single fluorine atom from acetonitrile and takes place in a solvent of acetonitrile. An example reaction from a molecular dynamics simulation is shown in Figure 3.8.

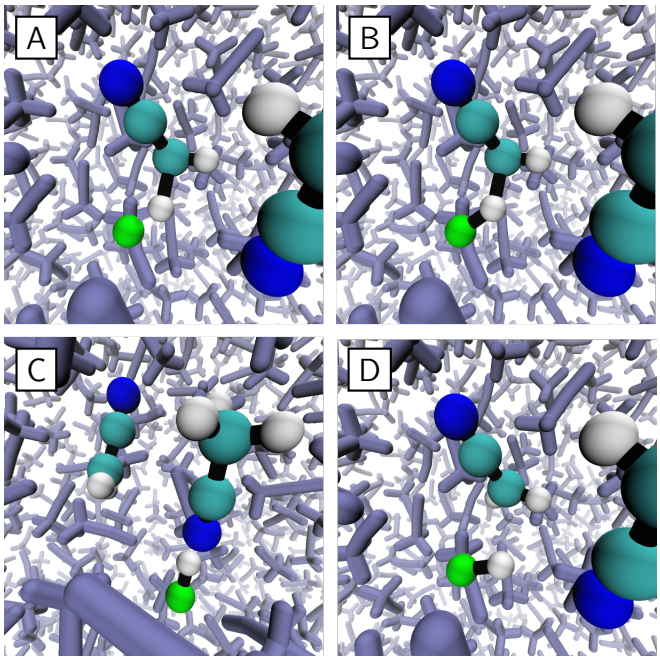


Figure 3.8: Snapshots from a molecular dynamics simulation of F + CD_3CN in an explicit solvent of 62 CD_3CN molecules. The images show: A) approach of F to a CD_3CN co-reactant, B) passage over the abstraction transition state, C) the nascent DF and its CD_2CN co-product, and D) formation of a hydrogen-bonded complex between DF and another solvent molecule. Source: Ref [131], used with permission.

3.4.1 Methods

The system is simulated using a custom version of the molecular dynamics package CHARMM (version c36a1) in which the multistate empirical valence bond method (MS-EVB) method is implemented[145]. The MS-EVB method provides a method for efficient simulation of reactive systems by modelling each user-defined valence state of the system simultaneously and coupling them together through a Hamiltonian matrix $H(\vec{r})$. The diagonal elements of this matrix $V(\vec{r})_1, \ldots, V(\vec{r})_n$ correspond to the energy of each state, and the off-diagonal elements are the coupling between each state. By diagonalising this matrix, the smallest eigenvalue-eigenvector pair can be extracted to give the energy of the system and the contribution of each state to the Hamiltonian, resulting in a smooth reactive potential. Further details of the method and a discussion of high-performance implementations of it which were published in Ref [45] are given in Appendix B.

In the previous work of Glowacki *et al* and what follows, the reaction was simulated in the solution phase through a reactive complex of a fluorine atom and an individual acetonitrile from which the deuterium could be abstracted, and a solvent of 62 additional acetonitrile molecules[6]. 64 EVB states were used to provide reactivity: one for the reactant state of F + CD_3CN, one for the product state of deuterium abstraction, and 62 additional states corresponding to hydrogen-bonding interactions between the produced DF and the nitrile groups on the solvent molecules.

In their molecular dynamics studies, BXD was used first used to equilibrate the system by restricting the distance between the F radical and the reactive deuterium between 1.5 Å and 1.8 Å, preventing the fluorine radical from diffusing away during equilibration. After equilibration, the lower bound of 1.5 Å was removed in the NVE runs, accelerating the rate at which the reaction occurred.

A free energy calculation of this reaction in equilibrium was not possible at the time due to the lack of support for multidimensional collective variables in the original BXD implementation. While restricting the F-D distance is sufficient to accelerate the sampling of abstraction, it is not enough for reversible reaction sampling. Once the reaction occurs, the FD product will diffuse away from the CD_2CN, at which point the F-D distance is meaningless in the context of accelerating the reverse reaction. To make sampling of the reaction reversible, an additional collective variable is required: the distance between the deuterium and the carbon atom from which it is abstracted. Constraining this collective variable prevents the product FD from diffusing away. With the new generalisations to BXD, the system could be sampled sufficiently to converge a free

energy calculation.

As well as the condensed phase, dynamics on the gas phase of the system was run for comparison, in which there are only 3 EVB states: the reactant F + CD_3CN state, the co-product DF + CD_2CN state, and the $[CD_3CND]^+$ + $[F]^-$ state. For both phases, the simulations were set up using the MS-EVB parametrisation as described in Ref [145], with a time step of 0.1 fs, a Langevin thermostat at 300K and a friction coefficient of 20 ps^{-1}, using Leapfrog integration.

The adaptive boundary generation procedure was applied to the system in both phases, with 100 ps of sampling between boundary placement in the gas phase, and 30 ps in the solution phase. The gas phase, consisting of only three states, was significantly faster to run and thus relatively long sample times were possible. A value of 0.01 was used for ϵ, the parameter controlling the placement of boundaries. This choice means a boundary was created based upon values of the collective variable $\vec{s} = (C-D, F-D)$ sampled 1% of the time.

Once the boundaries had been generated for both gas and solution phases, BXD dynamics were run on the system to converge statistics for free energy calculation. In the gas phase, a single 100 ns run was performed. As stated, the full 64 EVB state solution phase was significantly more computationally expensive to run, so the trivially parallel nature of the BXD algorithm was exploited to run dynamics in each box independently, totalling 12 ns of dynamics.

The MFPTs between each box was calculated as described in Eqn 3.1, and the box-to-box probabilities were computed as in Eqn 3.3. In the original BXD implementation, histogram binning is a straightforward partition of the single collective variable, enabling a fine-grained free energy profile of a single reaction coordinate ρ to be produced. The following method was used to define a reaction coordinate as a pathway through the hyperplanes of the multidimensional BXD implementation[1] (see Figure 3.12):

1. Define a path ρ which passes through the average point on each boundary (i.e. $\vec{s}_{max}$).

2. Bisect the box defined between two boundaries with additional hyperplanes up to a user-defined resolution (in the units of the CV space). For m bins to be placed between two boxes, $m-1$ bisections are computed as $\vec{n}_{bin} = (1-f/m) * \vec{n}_i + (f/m) * \vec{n}_{i+1}$, where $f \in \{1, 2, \ldots, m-1\}$.

[1]The full set of scripts for performing analysis of BXD trajectories are available at `https://github.com/mikeoconnor0308/bxd-analysis`.

3. Define the centre of each bin to be the point along ρ which is equidistant between the two hyperplanes defining it.

These histogram bins can then be used with Equation 3.4 and Equation 3.5 to calculate the free energy along the path ρ.

3.4.2 Results and Discussion

To illuminate the behaviour of the adaptive procedure for generating BXD boundaries, Figure 3.9 shows the BXD boundary generation procedure applied to the gas phase. The final set of bounds used for sampling are shown in Figure 3.10 for both the gas phase and solution phase.

The resulting time series and boundaries produced highlight several notable consequences of using an automated procedure. The size of the boxes placed differ depending on the underlying potential. The free energy surface up to the transition state (TS) is relatively flat, with a single boundary set at similar positions in both gas and solution phase. Once over the transition state, the produced DF rapidly diffuses away (increasing C-D distance) until it hits the boundary at 3.5 Å. From there, boundaries are placed in the other direction, heading towards a reversal of the reaction. As this progresses, a very steep landscape is encountered, resulting in several boundaries with relatively short box widths. Also of note is that the boundaries generally follow the dynamical pathway, and are placed orthogonally to it. Lastly, the solution phase results in 12 BXD boundaries compared to 10 in the gas phase. The difference in the number of boxes is due to a combination of two factors: friction and sampling time. In the solution phase, higher friction is experienced due to the solvent. As well as that, the relatively short sampling time of 30 picoseconds before placing a boundary means that values more distant from the previous boundary are less likely to be sampled, resulting in the boundary being positioned closer to the last. Effectively, this means the choice of sampling time before placing a boundary is proportional to the boundary size, and so can be chosen as appropriate to accelerate sampling. For the gas phase, broader BXD boxes are allowable since the dynamics were faster to compute, while in the solution phase smaller boundaries are desirable to increase acceleration, owing to the more expensive dynamics.

The boundaries generated with the adaptive procedure were then used to converge mean first passage times. The box-to-box mean first passage times for the solution phase are shown in Figure 3.11, with errors computed using block averaging (see Appendix C.1) to account for any correlation effects. Even with substantially less sampling than

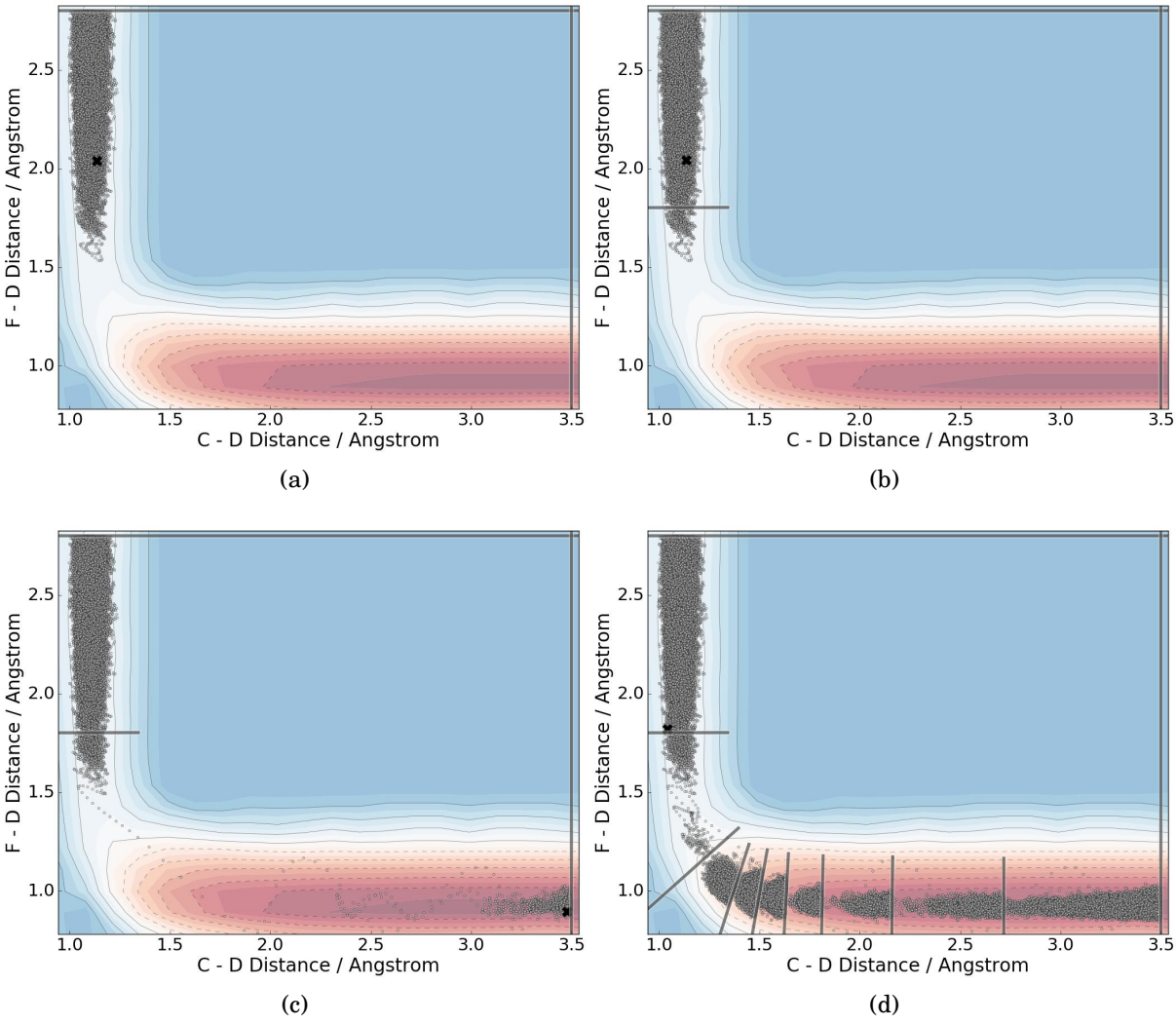
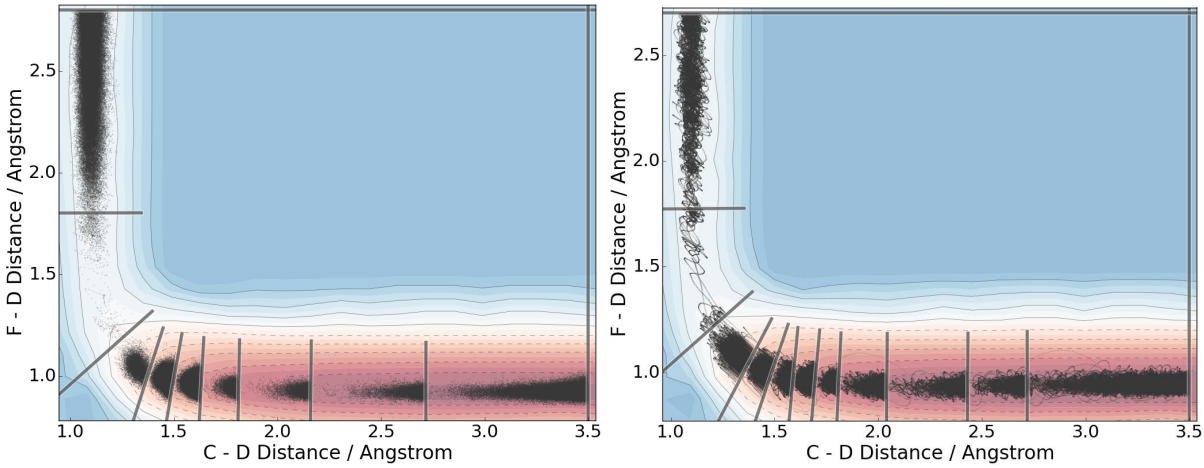


Figure 3.9: The adaptive boundary generation process for gas phase F + CD_3CN. The grey dots indicate points in CV space that have already been sampled, and the black X indicates the position of the system at the time when the snapshot for each respective panel was taken. The underlying potential energy surface is displayed for reference. Panel A shows initial sampling near B_R, and snapshot Panel B shows the generation of the first boundary. Panel C shows the state of the system immediately following transition state passage and rapid downhill transit toward B_R. Panel D shows the adaptive boundary placement as the system finds its way back to the first box (i.e., that which is bounded by B_R).



(a) Gas Phase

(b) Solution Phase

Figure 3.10: The BXD boundaries generated for both gas and solution phase F + CD_3CN. The grey dots indicate the sampled points in CV space, the grey lines represent the BXD boundaries, and the underlying potential energy surface is shown for reference.

the gas phase, the passage times are well converged, with the resulting box-to-box free energy profile also indicating convergence, as shown in Figure 3.11. Errors are calculated by standard propagation of uncertainty (see Appendix C.1) methods, and give an error of 0.15 kcal/mol.

The free energy calculated along the reaction path for the gas phase along with the histogram bins used to define the path reaction coordinate is shown in Figure 3.12. The figure shows how the free energy profile converges rapidly for this system.

The free energy profiles for both gas and solution phase are shown in Figure 3.13. The profiles are similar in the reactant phase and up to the transition state, with a slightly higher barrier in the gas phase. In the post-reaction region, however, the gas phase exhibits a deep well which is not present in the solution phase profile. In the gas phase, the DF product forms a hydrogen-bonding complex with the nitrogen of the CD_2CN, which corresponds to a C-D distance of approximately 4 Å. In the solution phase the product can form this complex with any of the solvent acetonitrile molecules; hence there are no particular minima evident in the profile.

It is worth noting the small discontinuity in the profile that is present in the gas phase profile at the value of ρ of approximately 2 Å. The discontinuity occurs in the third box (counting from an extended F-D distance). A plot of the sampled values of $\vec{s}$ in the third box in Figure 3.14a indicates that the region closest to the boundary is sampled less than the region slightly away from the boundary, which leads to the

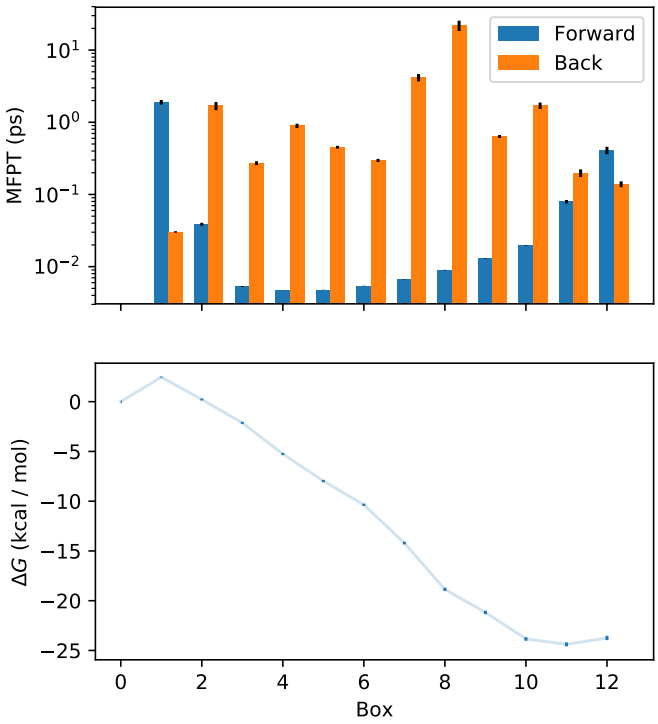


Figure 3.11: In the top panel, mean first passage times (MFPTs) for the solution phase between boxes are shown in each direction (blue from reactant to product, orange from product to reactant), with standard error bars computed through block averaging shown in black. In the bottom panel, the box-to-box free energy profile is shown, with error bars.

bin nearest the boundary having an increased free energy contribution. Intuitively, one would expect in this region of the free energy surface for an extended C-D distance to be favourable, and indeed this is observed clearly in the fifth box, shown in Figure 3.14b. It appears that the dynamics within the third box have a consistently ballistic nature in the region of the boundary, as shown for a short trajectory in Figure 3.14c. Because the surface is so steep, there is no time for the dynamics to decorrelate before another reflection against the boundary occurs. This ballistic nature leads to undersampling in the region very close to the boundary. Indeed, close inspection of Figure 3.14b shows that the area immediately to the left of the right-hand boundary in this box is also undersampled, but the effect is much smaller. For small boxes on such steep regions, the ballistic nature near the boundary has an impact on the histogram binning and

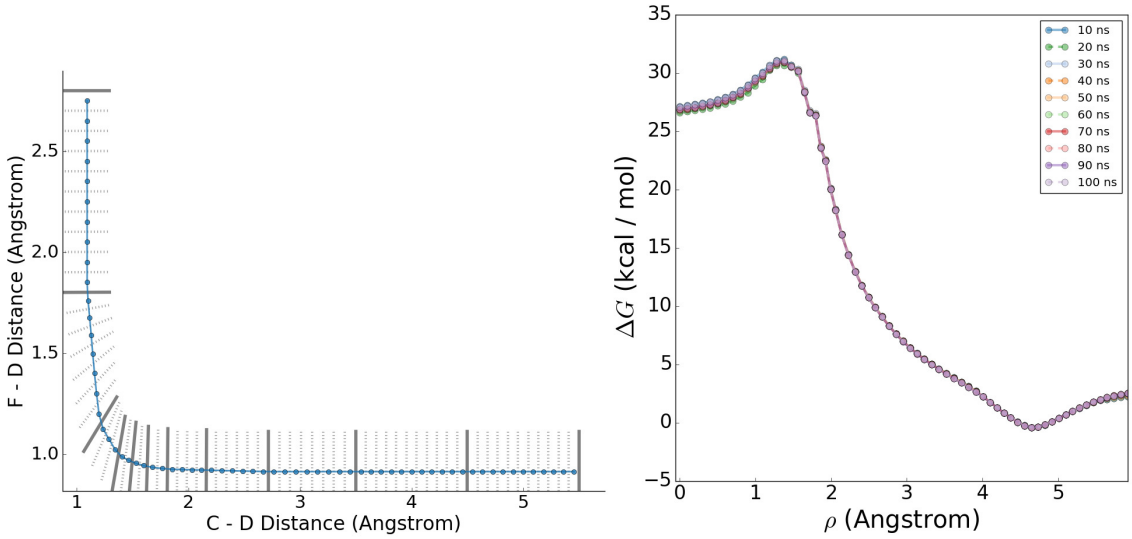


Figure 3.12: A) Illustration of the histogram binning procedure used to define ρ, the path from the reactant to product state, showing the BXD boundaries in dark grey, with dashed grey lines representing the bins. The blue circles are the centre of the bins, and the line between them represents the path ρ. B) Convergence of free energy as a function of sampling time in the gas phase.

final free energy surface as there are only a small number of bins in the box. Developing a robust method for accounting for these effects in the fine-grained histogram binning procedure remains a problem for future study, but they do not contribute significantly to the overall free energy profile.

For the purposes of validating the method, the free energy difference in solution phase between the transition state and the product represents an upper bound on the amount of energy that is available to the produced DF after the reaction. The observed value of 27.6 kcal/mol is in good agreement with the previous theoretical and computational studies in which 23 kcal/mol was found to be deposited into the stretching motion of the DF[6].

3.5 Conclusions

In this chapter, a generalisation of BXD to multidimensional collective variable space was presented, as well as an automated procedure for generating boundaries along a continuous path in collective variable space. The application of the new procedures to a reactive system that was not possible to study with the method previously, in which the

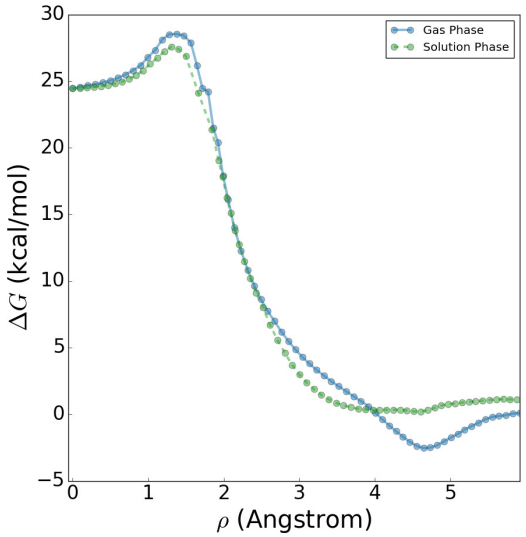


Figure 3.13: Free energy profile for gas and solution phase as a function of the projected reaction coordinate ρ.

results were in good agreement with previous computational and experimental studies, provides substantial evidence for the reliability and utility of the methods. The generalisations of the algorithm give it broader utility, as well as making it easier to fit into existing methodologies.

The methodology for finding paths and placing boundaries in the multidimensional space described has some shortcomings that could be improved upon. The first problem is a reformulation of the chronic problem that plagues collective variable based methods. The quality of the boundaries placed and the path that will be found depends strongly on the choice of collective variables. Implicit in the assumption of the boundary generation is that between the start and end points defined by the user there is a single path in the collective variable space that will be discovered by placing boundaries. If the collective variables do not adequately describe the transition, such a path may not exist. Similarly, if there are multiple viable paths or a bifurcation in the free energy surface leading to differing products, then the behaviour of the adaptive algorithm is not well defined. Requiring the user to correctly and precisely identify the set of relevant collective variables means that the method will still rely heavily on intuition and trial-and-error. However, there have been recent developments in using more general dimensionality reduction techniques to provide appropriate collective variables, such as the PCA and TICA algorithms[127], as well as machine learning methods such as autoencoders[61]. Alternatively, several descriptions of a path collective variable, where progress is defined along a series of configurations, have been described[146]. In these

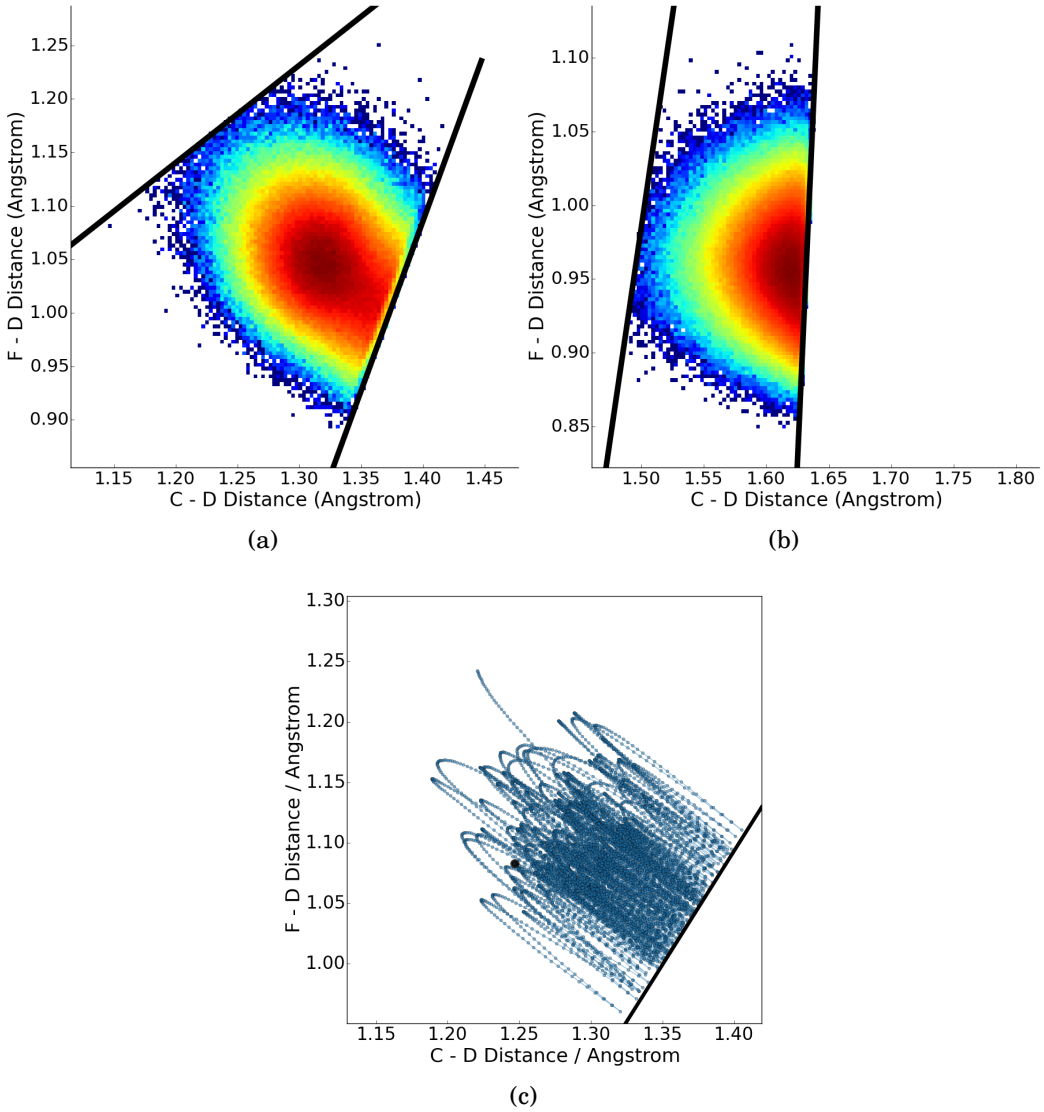


Figure 3.14: Histograms of sampled values of $\vec{s}$ in the third and fifth boxes of the F + CD_3CN system. The colour scale from blue to red indicates frequency of sampling. The third box is shown in panel A while the fifth box is shown in panel B. In panel C, a short trajectory in the region of the third box is shown, indicating the ballistic nature of the trajectories.

methods, initial data must be gathered which adequately describes the process. The algorithms presented in this chapter should be entirely compatible with such descriptions.

As previously observed, the algorithm for placing a new boundary based on the previous bound and observed dynamics is a greedy one, seeking to maximise the distance from the previous bound in the direction of the product. The critical assumption of this greedy method is that the direction in collective variable space that exhibits maximal variance, i.e., the local path of least resistance, is the globally relevant pathway in the region of collective variable space defined by the user. Equivalently, the method always proceeds along the pathway that is locally the lowest free-energy path in collective variable space. While pathological cases exist in which this approach will fail, it is not immediately clear how common such failures will be. The algorithm could be improved by allowing the BXD algorithm to explore multiple pathways simultaneously when directionality is not apparent. Such an extension would also enable accelerated sampling of bifurcating reaction pathways. Alternatively, one could use path collective variables[129] (discussed in detail in Chapter 6) to provide an initial guiding path which could be followed.

For the BXD method to become more widely used, further tests to its robustness and methods for calculating error and convergence criteria will need to be developed. The introduction of block error analysis to reduce correlation effects in the calculation of errors in the MFPTs and propagation of error through to the box free energies mark some progress in this regard, but further rigour is required. In particular, methods for validating and assessing the convergence of the fine-grained free energy profile produced through histogram binning have not yet been developed. The bootstrapping method, in which existing data is sampled with replacement to allow confidence intervals to be calculated, is a promising avenue as it avoids the need for complex propagation of uncertainty calculations[147].

The focus of the development described in this document was for accelerated sampling along a particular reaction pathway which could be well defined by an appropriate choice of collective variables. However, the methods developed can be used in a broader context. In particular, the method has been further extended for use in accelerated sampling of reaction networks[124]. Here, the potential energy of the system was used as the reaction coordinate to encourage the system to undergo reactions, utilising the velocity inversion procedure described in this thesis to restrain the system. Because the goal in this application was to map configuration space rather than to converge statistics, the

adaptive procedure was altered to allow the system to continuously explore new regions of phase space, by adding and removing barriers on the fly. This novel development to BXD demonstrates the flexibility of the method, which stems from the simplicity of the original algorithm.

CHAPTER 4

INTERACTIVE MOLECULAR DYNAMICS

The use of BXD to accelerate the sampling of rare events such as those described in the previous chapter requires the user to determine a few key collective variables before use. This requirement is shared among many of the most commonly used rare event acceleration methods discussed in Chapter 2. Determining these collective variables is a highly non-trivial task that requires a mixture of chemical intuition and a significant amount of trial and error. For simple systems such as the F+CD_3CN reaction previously studied, this is rather easy (but still time-consuming to identify). For substantial conformational changes such as those found in protein folding or drug-binding, it can become extremely difficult to identify these collective variables. If a poor choice of collective variable is made, rare event methods will usually fail, encountering hidden barriers or entering irrelevant regions of phase space.

An example of this was seen in the previous chapter, where using the F — D distance alone was insufficient for the F+CD_3CN system, necessitating the development of an extension to BXD to multidimensional collective variables. As discussed in Chapter 2, there is always a trade-off between spending time identifying key collective variables that enable the use of efficient accelerated molecular dynamics methods and using methods that require less information but more compute power. While developments in machine learning approaches are beginning to attempt to automate the process of extracting collective variables[128, 130], in many cases, it seems for now at least human intuition will still play a role in identifying important dynamical pathways to study. An enticing method for systematically introducing human intuition into the scientific work-

flow for accelerating molecular dynamics is through interactive molecular dynamics. By introducing interactive biasing potentials into a molecular dynamics simulation, a user can accelerate processes of interest, and use the resulting structures and pathways as starting points for further analysis.

In the following chapter, I present a brief history of interactive molecular simulation and virtual reality technologies and present a framework for interactive molecular dynamics using commodity virtual reality that can run on remote computer infrastructures. This work builds my previous experience in developing GPU-accelerated interactive molecular dynamics using commodity depth-sensors[11, 148]. The framework and utility of VR for molecular tasks are then evaluated with a user study in Chapter 5, which resulted in a publication in Science Advances[149]. My contributions to this work were in the development of a modular framework for interactive molecular dynamics, implementation of molecular dynamics algorithms, and design and implementation of the virtual reality experience. The framework was developed in collaboration with developers at Interactive Scientific Ltd, who provided support, funding through a CASE studentship, and designed and implemented the cloud architecture, the communication layer, and the molecular renderers. The input files and resulting datasets for all simulations and experiments described in this chapter are available in Ref [150].

The idea of interactive simulations goes back to work carried out in the late 1980s by Fred Brooks with their visionary Project GROPEHaptic[1], in which a room-sized haptic device combined with a 2D screen was used for molecular docking tasks[151]. In a user study, they found that binding poses found interactively were better than those of the algorithms used at the time and that haptic feedback improved user performance. It is worth noting that due to computational power available at that time, energy and forces were precomputed in a grid across the relevant configuration space, so users had limited exploration options. At the same time, the group developed an interactive protein manipulation tool, which allowed users to manipulate protein structures with real-time minimization[152].

As computing power increased exponentially and graphical processing developed, haptic devices were miniaturised, 3D visualisation hardware became available, and molecular simulations became much faster, and so the idea was developed further by Schulten and co-workers in the early 2000s[153, 154]. It was now possible to run full molecular dynamics simulations interactively in 'real-time'. Here, real-time means that each time step of the simulation is calculated fast enough so that the results can be

[1] An unfortunate name.

displayed to the user at a responsive speed - the simulation itself is still simulating dynamics at a rate orders of magnitude slower than 'real-time'. An interactive form of Steered Molecular Dynamics (SMD) was implemented, with the term Interactive Molecular Dynamics (IMD) being coined. Unlike SMD, in interactive molecular dynamics, a user could vary the application of biasing forces throughout the simulation, allowing more delicate control. This fundamental difference means that a user can begin an interactive session without having to define any specific distances or collective variables to steer, instead, they can explore hypotheses on-the-fly as they use the biasing potentials to steer the simulation towards desired events.

The system connected the existing visualisation tool VMD with the molecular dynamics package NAMD over a TCP/IP connection, with a spring force applied to atoms that could be felt in the haptic feedback device. The molecular dynamics package transmits atomic positions every few steps (a variable controlled by the user) to the visualisation client, and the client returns an array of interactive forces to be applied to atoms, which are integrated into the Newtonian mechanics of the system.

Schulten's group performed some applications with their IMD system, the first being exploration of conduction pathways of the monosaccharides ribitol and arabitol through the membrane channel GlpF[155]. IMD was used to perform initial steering of the molecules through channels, providing qualitative insight. Of particular note is that the unique haptic interface itself provides insight: "as the molecule of ribitol or arabitol is pulled into the channel, a strong resistance is felt in the user's hand, and he must push forcefully to overcome it ... after the sugar molecule has been pulled 10 Å through the channel, the resistance lessens noticeably"[155].

As the authors note, the interactive molecular dynamics sessions provide only qualitative information, so follow-up dynamics was performed at various snapshots produced in the interactive molecular dynamics. These short simulations were used to equilibrate the various configurations found with interactive molecular dynamics, but an unbiased characterisation of the pathway is not given.

In another application, investigating anionic conduction in CIC channels, the IMD system was used to manipulate a structure before follow-up simulation[156]. Finding that the relevant pore was blocked due to the orientation of a glutamic acid residue (Glu148), IMD was used to adjust the position of the residue such that the pore became open. While the exact conformational change the protein must undertake to allow ion transport was not known, it was reasonably assumed that the Glu148 residue would have to be moved. With the new conformation created, umbrella sampling was used

to perform accelerated sampling of the conduction process. Similarly, a more recent application used interactive molecular dynamics to adjust initial configurations in RNA simulations[157].

Other groups developed their own implementations, for different applications. A generalisation of the method with the development of a library for communicating with MD packages, as well as application to coarse-grained simulations was developed by Baaden and colleagues[158]. Meanwhile, at the atomistic level Dreher *et al* developed a high-performance, scalable framework, resulting in an impressive 1.7 million atoms that could be simulated and visualised with molecular mechanics at interactive speeds[159].

As increasing computational power has enabled simulations of hundreds of thousands of atoms to be simulated and interacted with using molecular mechanics force fields, likewise accelerations in electronic structure calculations mean that interactive quantum chemistry has now become feasible. Reiher and colleagues use semi-empirical methods combined with continuous minimisation to allow interactive exploration of the potential energy surface of reactions, enabling reaction pathway discovery[160–162]. In another approach, Martinez and colleagues' development of GPU-accelerated quantum chemistry enabled the VMD IMD framework to be applied to *ab initio* molecular dynamics, resulting in the ability to accelerate the exploration of chemical reactions such as proton transfer[163].

Arguably the most high profile interactive molecular simulation application is Foldit, by Cooper and colleagues [164–166]. This application is a more specific form of molecular simulation applied to protein structure prediction and does not use atomistic molecular dynamics methods. Instead, the application is an interface to the Rosetta protein structure prediction tools, wrapped in a user-friendly gamification layer allowing users to perform specific actions upon protein structures to change their structure. These actions include pulling on sections of the protein, rotating helices and introducing constraints, as well as applying simplified algorithms from the Rosetta tools such as "combinatorial side-chain rotamer packing ('shake'), gradient-based minimisation ('wiggle'), and fragment insertion ('rebuild')"[167]. These actions are based on the algorithms used by automated methods for searching the configuration space of protein folding. This application can be thought of as a step-by-step puzzle, in which the user is deciding which of the tools to apply next, in the same way that a stochastic optimisation method would decide whether to apply a minimisation or attempt to escape minima. Therefore, despite not having a 3D interface, users were able to build 'recipes' of actions that successfully

solved the puzzles, finding folded states. Of particular note is the crowd-sourced, cooperative/competitive nature of this application, in which a web interface enabled users to share scores and strategies, resulting in collective improvements to the approaches used and the resulting protein structures. As these strategies spread and became engrained into the community, it was discovered that the algorithm within two of these recipes was similar to novel algorithms being developed by Rosetta developers at the time and that they outperformed existing automated algorithms[167]. This was a demonstration that interactive simulation not only produces results in applications but can also improve existing automated methods.

Given the success of Foldit, it is worth considering whether interactive simulation can be successful in broader applications, such as drug binding, protein dynamics and chemical reactions, and more broadly as a tool to assist molecular simulation workflows. Despite several implementations of interactive molecular dynamics (IMD), there have been few applications (the most notable being described above) and relatively little adoption of IMD into the computational chemistry workflow. The main reasons for this are the size of simulation that can be routinely simulated, the problem of simulation timescales, and the availability, cost and quality of interaction and display equipment. However, developments in both hardware, software and algorithms suggest there may be some remedies to these issues and are discussed in turn.

Simulation Scaling and Timescales

The first limitation is the size of simulation that can be simulated and visualised in real-time. While the systems described above can now be used with large systems[159], the hardware required has not previously been generally available. The need to use shared HPC clusters for performing simulations does not lend itself to an interactive workflow. Simulations are usually submitted in batches to a queue system, making it difficult for existing IT infrastructures to provide interactive sessions. However, the availability of low-cost GPUs and the development of GPU-accelerated molecular simulation packages means that systems of an interesting size can now be simulated routinely on a workstation, expanding the scope for interactive simulation. Additionally, the rise of containerised and cloud-based compute resources has increased the availability of affordable on-demand compute. An example of this is in Jupyter notebooks, in which simulations are instantiated, run and analysed within an isolated environment[168].

A related problem is the timescales of molecular simulation. All-atom molecular mechanics simulations use time steps at the scale of femtoseconds, while rare events

occur on the timescales of picoseconds, microseconds or even milliseconds depending on the system and the event, as discussed in Chapter 2. From the user's perspective, this means that a minute spent in an interactive session corresponds to the order of a few picoseconds of simulation time. In order to accelerate events within a reasonable interactive session, one typically needs to apply strong interactive forces, which can result in unphysical pathways.

Furthermore, applying such strong forces results in a significant bias, so results are primarily qualitative or form the starting point for further study. The impact of this depends on the application. If interactive simulations are used to as a tool to simplify the process of setting up simulations, for example to position molecules near a site of interest as in Ref [157], then this is not an issue. To be able to use IMD to truly accelerate processes, however, pathways generated with IMD need to be combined with additional sampling methods. The real value of an IMD session is as an efficient method for expressing a user's intuition for a potential pathway, rather than a precise definition of it.

As described in Chapters 2 and 3, in recent years there have been many developments of accelerated sampling methods that require initial pathways, configurations or the identification of collective variables or features for dimensionality reduction. For example, linear interpolation is used regularly to initialise a nudged elastic band[169, 170] (NEB) or path metadynamics simulation[129], but linear interpolations can often produce physically unrealistic pathways that can be challenging to optimise[171]. Another approach is to use biased simulations to produce an initial pathway. It seems that combining interactive molecular dynamics with an appropriate sampling method could address the problem of timescales and bias, while simultaneously providing a way of combining human intuition and automated method to produce statistically significant results. A high-level proposed workflow is given in Figure 4.1. The workflow is intentionally non-specific, allowing for it to be used in combination with any of the myriad of existing methods, or 'module', used in computational chemistry. Given paths or configurations generated in IMD, these are passed to some method for feature extraction. This may be the identification of collective variables for a method such as BXD, the optimisation of a path for methods such as NEB or path metadynamics, or the extraction of features for the building of a Markov state model. These features are then passed, along with the relevant initial conditions, to the appropriate sampling algorithm, which converges statistics to produce the desired observable, which may be free energies or rates. Ideally, the steps between an IMD session and the observable ought to be automated as

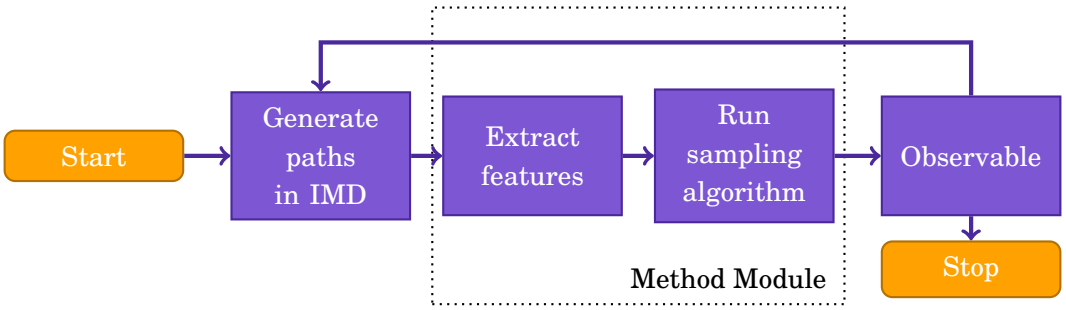


Figure 4.1: A high level illustration of a proposed workflow for interactive molecular dynamics. The dotted bordered region represents an application specific method.

much as possible, resulting in a black-box method from intuition to quantitative results. Given the result of an observable calculation, a user may return to IMD to generate more hypotheses, using the information gained about the system.

3D User Interfaces and Virtual Reality

As well as the limited availability of hardware for simulations, similarly the availability of graphical workstations, 3D displays and the haptic devices used to control and visualise the simulations in the implementations described were a barrier. As well as being expensive, these setups do not provide an intuitive manner in which to control three-dimensional objects such as molecules. 3D displays provide depth information, but the user's viewpoint is mostly fixed, and they must rotate the camera to observe and manipulate the system at different angles. While haptic devices such as the Phantom Desktop do allow for 3D control[153], the range of movement is limited, and typically not co-located. Co-location is the experience of input being located where it is being displayed. For example, using a touch-screen to draw on a 2D image is co-located, while using a mouse to achieve the same task is not. In a 3D environment using haptic devices such as the Phantom, it is difficult to achieve co-location. However, when it is achieved, user ability to perform tasks has been shown to improve[172]. This lack of true 3D control is reflected in the set of applications that IMD has been used for: tasks like the transport of ligands and ions through channels are relatively simple pathways to explore, and so can be achieved with limited 3D manipulation.

The 3D-display and haptic devices used in previous attempts at IMD could be considered a particular implementation of virtual reality, and recent developments of virtual reality technology may address the problem of 3D visualisation and manipulation with greater success. The term virtual reality (VR) is used to describe many technolo-

gies and experiences; VR pioneer Fred Brooks' defines a virtual reality experience as "any in which the user is effectively immersed in a responsive virtual world"[173]. Virtual reality has been a technological frontier in development since the 1960s, with Ivan Sutherland being one of the first to consider what such a human-computer experience would be like and to develop the first head mounted display (HMD)[174, 175]. Indeed, Sutherland considered that such a display would not "have to follow the ordinary rules of physical reality with which we are familiar", a prescient statement as we consider its utility for understanding the physics governing molecular systems. The 1980s saw the first wave of public enthusiasm for virtual reality, as Jaron Lanier, a pioneer in VR, formed the company VPL that sold the first commercially available virtual reality devices. The technology of the time did not live up to its promises, and the idea appeared to 'die'[176]. However, in a 1999 survey by Fred Brooks of the use of VR in industry, and a more recent review by Slater in 2014, it was observed that the technology had been quietly adopted by industry and was continuing to be developed[173, 177], particularly in medical applications[178]. Numerous technologies including HMDs, body tracking and CAVE systems are in use.

Today, driven by the consumer video games market, advances in graphics processing, the miniaturisation of displays and sensors have led to a seemingly sudden commoditization of virtual reality (VR) hardware and another wave of public interest. In particular, head-mounted displays (HMDs) have become synonymous with VR, and there has been a proliferation of solutions and terms including 360 videos, mixed reality (MR), and augmented reality (AR) that are generally encompassed under the umbrella term of extended reality (XR). It is worth distinguishing between some of these technologies, as the possibilities for immersion, visualisation and control are quite different.

At the time of writing, the state-of-the-art in commodity VR is the HTC Vive and Oculus Rift[179], in which a head-mounted display covers a user's field of view, replacing the physical world with a virtual world. The technologies combine optical tracking and inertial movement units to provide full 6-degree of freedom movement and tracking to within one centimetre accuracy within a space up to ten by ten meters. Dedicated controllers, reminiscent of video game controllers with numerous triggers and buttons, are also tracked in the virtual space and so provide a co-located link between the user and the virtual world: they see a virtual representation of the controllers in the virtual world that matches the location of the controllers in the physical world. Thus these systems provide solutions to 3D co-location problems and provide immersive 3D experiences and interaction.

At the low-budget end of virtual reality technologies, a mobile phone is placed in stereoscopic goggles. By relying solely on the AMUs of mobile phones or the goggles the experience is limited to the rotational degrees of freedom, and without tracking there is no way to interact with what is being seen in 3D. While providing an accessible entry point into virtual reality, VR pioneers such as Lanier make the point to distinguish between such technologies from the higher-end offerings: "If you can't reach out and touch the virtual world and do something to it, you are a second-class citizen within it... a subordinate ghost that cannot even haunt"[180].

While it remains to be seen whether this wave of enthusiasm for consumer virtual reality will be short-lived, the level of immersion achieved with modern (high-end) virtual reality, and the resulting feeling of *presence* - that one is actually temporarily inhabiting the virtual reality - is highly convincing[181]. This immersion and the enabling of true 3D visualisation has resulted in applications of VR becoming more common in industry and research[176], suggesting that it may be here to stay.

Augmented reality (AR) / mixed reality (MR) distinguishes itself from VR by overlaying virtual elements over the physical environment. Combining the virtual and physical environments has the advantage of not isolating the user from the physical world and others. Additionally, AR/MR is often developed as a self-contained system, and so the tangle of wires currently affecting virtual reality does not apply[2]. However, a number of distinct challenges remain to be addressed convincingly.

The most difficult challenge to overcome that is distinct from VR is the requirement for the system to have a semantic understanding of the physical world, in order to display virtual objects within it convincingly[182]. Additionally, the considerations of latency, graphics quality, and interaction options, must be solved again, as the self-contained systems have limited computational capability when compared to the workstations used for VR. In the same way that there is a wide variety of VR hardware to accommodate different budgets, the quality of the experience in the various AR implementations currently available vary significantly. These technologies are still in a nascent stage of development, but the potential applications make them an exciting technology to follow in the years to come.

With virtual reality proving to be a valuable tool in many domains, particularly those that depend upon 3D reasoning such as surgery, computer-aided design and architecture[173, 176], it seems appropriate to determine whether the understanding and manipulation of molecular systems would be enhanced with virtual reality. Several VR

[2]Wireless VR solutions are now available, but in early stages of adoption.

applications for chemistry have emerged, so far mostly porting molecular dynamics visualisation and analysis tools to virtual reality[183–186]. With my existing experience in developing interactive molecular dynamics with novel control schemes[11, 148], it was deemed to be worth revisiting the concept of interactive molecular dynamics, with a particular emphasis on using it produce initial guesses and pathways with which one can accelerate the study of rare events. At the time of development, the experience provided by the HTC Vive seemed the most promising technology for enabling intuitive, co-located manipulation of molecular systems. In the following sections, the necessary system architecture, algorithms and user experience considerations for interactive molecular simulation in virtual reality are presented.

4.1 A Platform for Interactive Simulations

The open-source framework for interactive molecular dynamics in a virtual reality environment presented in this section began as part of the Nano Simbox platform, growing out of the aforementioned interactive molecular dynamics implementation[11]. It was developed in collaboration with Interactive Scientific Limited as part of my CASE PhD, and provides a modular and scalable framework for running, visualising and interacting with simulations. The open-source release of the software is, at the time of writing, referred to as NarupaXR[3]. The name Narupa is a portmanteau of the words nano and arūpa, a Sanskrit word referring to formless, or non-material objects. In the remainder of this document, the phrase iMD-VR will be used to refer to the software.

The iMD-VR architecture consists of a few fundamental concepts upon which specific applications can be built. The core concept is a client/server model of visualisation and computation, in which computation and hosting of simulations and content are performed on a server, which may be deployed locally on a client machine, on a remote cluster or a cloud-based compute resource. Client applications connect to this server to perform visualisation and interaction, as shown for a VR interface in Figure 4.2. This connection takes place over a simple TCP/IP connection if the server is on an on-site network, or over a WebSocket if connecting to remote or cloud-based computing resources. The WebSocket implementation provides scalability, as it is easier to use with existing IT infrastructures and enables the use of well-established web technologies for authentication, security and routing. The client/server design, as well as separating visualisation from physics computation, provides another advantage in that it is easy to

[3]Available at https://gitlab.com/intangiblerealities.

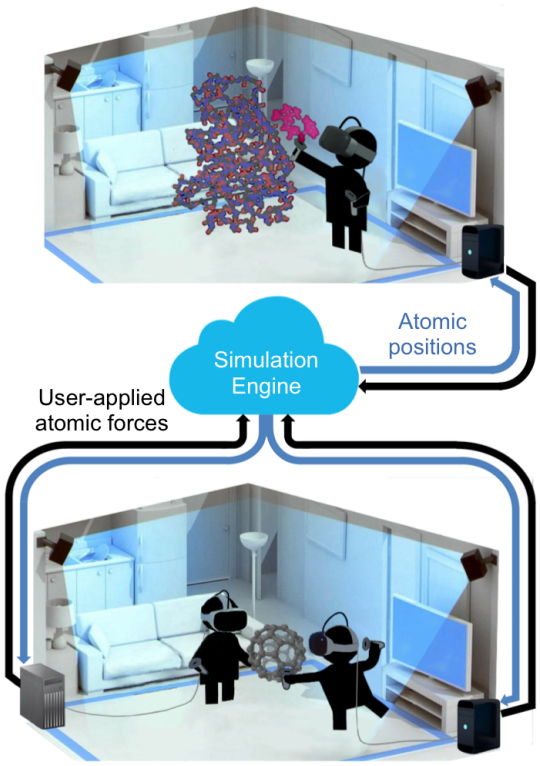


Figure 4.2: Schematic of the HTC Vive VR set-up of iMD-VR. The bottom panel shows two users within the multi-person VR framework passing a simulated buckminsterfullerene molecule back and forth. Each user's position is determined using a real-time optical tracking system composed of synchronised IR light sources. Each user's HMD is rendered locally; molecular dynamics calculations and maintenance of global user position data take place on a separate server, which can be cloud-mounted. The top panel shows a single-person set-up, where the user is chaperoning a real-time GPU-accelerated MD simulation to generate an association pathway that docks a benzylpenicillin ligand (magenta) into a binding pose on the TEM-1 β-lactamase enzyme. Source: Ref [149] used with permission under Creative Commons Attribution-Share Alike 3.0 Unported license.

adapt to produce a multi-user experience. For teaching, sharing results and collaborating on complex problems, the ability to visualise and interact with the same simulation can be extremely beneficial.

Data between the client and server is sent in two manners. Bulk data is transmitted using *streams*, which are capable of transmitting arrays of fixed length data structures using a simple TCP/IP socket implementation. These streams are intended to be used with simulation data that changes regularly, such as atomic positions, or interactive forces. Control structures provide a simple interface with which to query the current state of a stream and to read/write to it.

Data that does not fit into the requirements of the bulk data streams are communicated using the Open Sound Control protocol[187]. This protocol provides a URL-style address scheme, meaning that specific data can be associated with an address. Clients can subscribe to these addresses to trigger methods upon the receipt of a message matching the given address. This can be used in a wide variety of scenarios, such as transmitting the state of a system variable, or commands from clients to the server. For example, the command to pause a simulation is simply */pause*, while the command to resume playback is '*/play, 30*', where the argument specifies the frame rate at which to play.

With this client/server model and flexible communication protocol, specific applications can be built rapidly. While the work presented in this thesis is dedicated to molecular simulation, and what follows is an overview of the modular design strategy for interactive molecular dynamics simulations, it should be noted that the framework described is general can be used to visualise other types of simulation and data.

The molecular dynamics implementation in the iMD-VR framework makes heavy use of object-oriented design concepts. A server instantiates a simulation object, which is a base class for molecular simulation. This consists of an atomic system object and an integrator interface. The base atomic system object, in turn, consists of a topology describing the grouping of atoms and the bonds between them; the current position, velocities and forces of the atoms; a set of force fields that determine how the atoms interact; and any thermostats or barostats. By explicitly designing with modular components in mind, as displayed in Figure 4.3, the simulations are highly flexible. There are many integration schemes, thermostats and force fields used in molecular dynamics simulations, and so it is beneficial to expect customisation to take place.

Furthermore, writing an efficient and accurate force field that performs well on modern parallel architectures is not trivial. By actively designing in a modular fashion with a simple API and specifying only interfaces for what a simulation or force field must provide, it becomes straightforward to interface the iMD-VR framework with existing force field implementations and molecular dynamics programs, avoiding duplicate work. A plug-in scheme exists, allowing such additional functionality to be developed without recompiling the whole program, which enables users to customise their installation, and encourages experimentation and extension while maintaining a sustainable code base.

As a starting point, a relatively simple set of molecular dynamics algorithms were developed, including velocity verlet integration, a serial MM3 force field implementation[188–190] and a Berendsen and Andersen thermostat, based on implementations

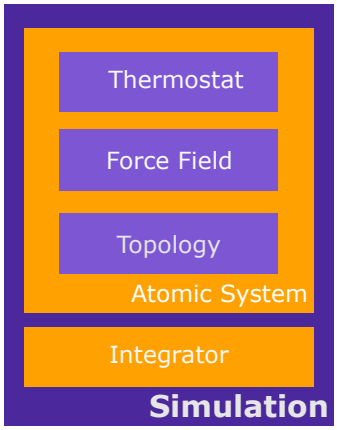


Figure 4.3: The iMD-VR Molecular Dynamics Architecture.

of Glowacki and co-workers[11]. These implementations serve as references for future development and enable basic testing of the architecture stack. Additionally, several plug-ins have been developed which integrate the functionality of existing codes. For classical molecular mechanics, a plug-in with the OpenMM molecular dynamics package[191], which provides GPU-accelerated support for a wide variety of force fields including Amber, CHARMM and various implicit and explicit water models, hugely extends the simulations that one can run interactively. In this plug-in, an implementation of the CCMA and SETTLE constraint algorithms[141, 142] are also implemented, based on the OpenMM implementation[142, 191], enabling simulations that make use of fixed bond lengths.

To enable molecular dynamics simulations of chemical reactions, plug-ins with DFTB+, an implementation of the DFTB method[29], and the semi-empirical force fields developed by Reiher and co-workers, including DFTB and PM6, have been produced[160–162]. So far, it has been found to typically be more natural to interface at the force field level and perform the integration internally, so that considerations for interactive simulation such as interactive biasing potentials and visualisation can be preserved.

In some applications, however, it can be advantageous to allow an external molecular dynamics program to set up and integrate the simulation. This may be for a few reasons: to enable rapid prototyping; because the desired functionality of the program is not easily incorporated into a modular component; for performance reasons; or because one does not wish to interrupt their workflow to run an interactive simulation.

The VMD IMD API, previously described, provides a way to achieve this[153]. In the API, a TCP/IP connection is established between a client and the MD program, and the MD program transmits coordinates and basic information such as the energy of the

system while the client returns interactive forces to be applied to the system. An implementation of the VMD IMD API was available in the PLUMED package, which is well suited to the task as it is incorporated into many existing MD programs to apply biases in the form of atomic forces. The implementation was removed in the official PLUMED v2.2 distribution, but it has been re-enabled in a fork maintained by the author[4] and developed further by adding support for pausing the simulation, adding transmission of the periodic boundaries of the system, and adding the ability to transmit a subset of the atoms in the system to improve performance. Furthermore, the communication protocol has been made more robust with platform/language independent serialisation of atom positions and forces. A corresponding C# implementation was implemented for use with the iMD-VR APIs.

The protocol consists of a socket connection between a client that provides the interactive forces and a server that provides the position of the atoms and other simulation data. Data is transmitted as a 16-bit header followed by a payload of data. The 16-bit header specifies the type of data being transmitted (atom positions, forces, periodic bounds, simulation data, or control signals) and the length of the payload. The simulation data consists of a data structure including the time step, the temperature of the system, and the energy of the system.

The implementation serves as an example of the flexibility of the iMD-VR framework, as the entire Simulation module is replaced with a VMD.IMD component, as shown in Figure 4.4. In this context, the simulation details are handled by the molecular dynamics package, while the iMD-VR framework provides visualisation, handles interactions, provides topology awareness, cloud compute support, and a shared session for multiple clients (MP Data in the figure).

Incorporating this way of running interactive molecular dynamics simulations through PLUMED enables an interface with a wide variety of programs, including OpenMM, Gromacs, Amber14, LAMMPS and DL-POLY to name a few. The integration is a balance between generality and functionality. It allows an interactive session with many of the popular molecular dynamics programs, but in order to be compatible with so many different packages and be minimally invasive, it has very limited functionality. A client can only apply interactive forces to atoms and query the basic information detailed above. Additionally, the architecture introduces two transfers of atom positions across the network. The VMD IMD API provides an avenue for rapid evaluation of interactive simulations with particular applications, after which the modular design of the iMD-VR

[4] In a fork available at `https://github.com/mikeoconnor0308/plumed2`.

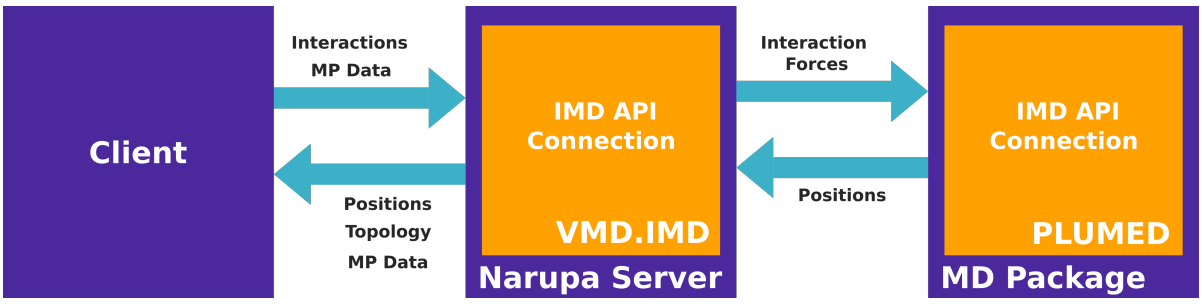


Figure 4.4: Schematic illustrating how a generic connection to existing molecular dynamics codes for the purposes of interactive molecular dynamics can be achieved with the VMD IMD API.

framework can be leveraged to develop specific plug-ins for particular applications.

4.2 A Virtual Reality Environment for Interactive Molecular Dynamics

The flexible nature of the iMD-VR platform and an existing basic molecular visualisation application developed in Unity3D in collaboration with Interactive Scientific provided a good starting point for virtual reality developments. As described above, the HTC Vive virtual reality hardware is at the time of writing one of the best consumer grade solutions available, and so the VR client was designed for use with it. It should be noted, however, that by developing the client in Unity3D, it is straightforward to port to other technologies as they emerge.

With the HTC Vive hardware, users are tracked within a space of up to 10 m x 10 m, allowing them to move around freely[192]. This particular feature has significant implications in the design of the user experience for molecular simulation. Since the user can move freely around the space, the simulation can be implemented as an object within the space that the user can move, rotate and rescale: a virtual analogue of the physical models familiar to many chemists. Users interact with the molecular simulation through a pair of controllers that are tracked within the virtual environment, enabling them to reach into the simulation and interact with the model intuitively.

In the following sections, algorithms and user experience considerations for performing interactive molecular dynamics in virtual reality are presented.

4.2.1 Interactive Potentials

Pulling the trigger on a controller in a 'live' simulation - one that is running real-time molecular dynamics - results in an interactive biasing potential being applied, as depicted in Figure 4.5a. As described in Chapter 2, the atomic motion is simulated by integrating the set of forces acting on atoms, which come from the potential energy of the system via

$$\vec{f}(t) = -\frac{\mathrm{d}V}{\mathrm{d}\vec{r}}$$

where $\vec{r} \in \mathbb{R}^{3N}$ is the vector of atom positions for a system of N atoms. In the interactive molecular dynamics system, an external component to the potential energy is introduced in the form of the user's interactive forces, splitting V into the components

$$V = V_{int} + V_{ext},$$

where V_{int} corresponds to the internal energy of the system, while V_{ext} corresponds to the potential energy a user exerts upon the system when interacting with it. The forces acting on the atoms then become

$$\vec{f}(t) = -\frac{\mathrm{d}V_{int}}{\mathrm{d}\vec{r}} - \frac{\mathrm{d}V_{ext}}{\mathrm{d}\vec{r}}.$$

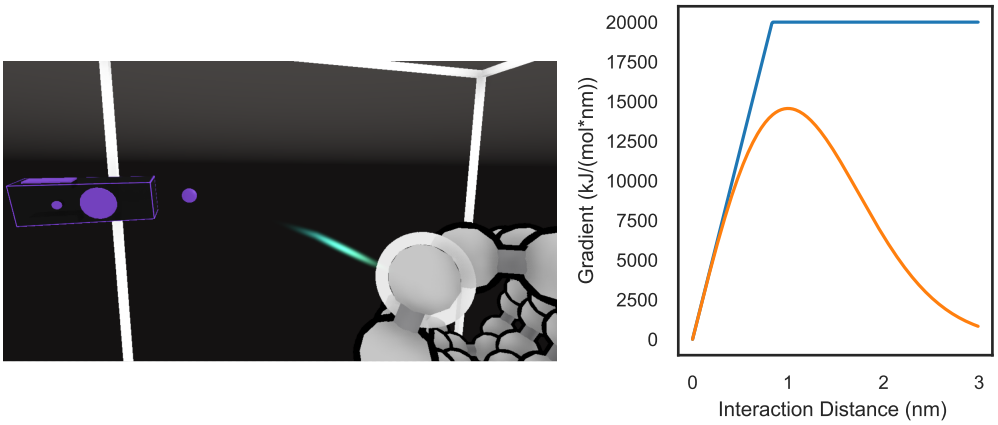
The user can interact with atoms in the system with two different potential forms. The first is implemented by projecting a spherical Gaussian field into the system at the 3D location specified by the user, $\vec{g}_i$, which acts upon the nearest atom j via

$$\frac{\mathrm{d}V_{ext}}{\mathrm{d}\vec{r}_j} = \frac{m_j c}{\sigma^2}\left(\vec{r}_j - \vec{g}_i\right)\exp\left(\frac{-\left\|\vec{r}_j - \vec{g}_i\right\|^2}{2\sigma^2}\right),$$

where c is a scale factor that tunes the strength of the interaction, m_j is the mass of the selected atom, and σ controls the width of the interactive Gaussian field. The value c can be tuned on the fly by the user, allowing them to adjust the required strength of the interaction for a given task. This formulation is a generalisation of the form used in the previous interactive molecular dynamics system to make it applicable to a 3D virtual environment[11].

An alternative potential is a spring force, which has been used in previous iMD frameworks such as the VMD implementation[193], and takes the following form:

$$\frac{\mathrm{d}V_{ext}}{\mathrm{d}\vec{r}_j} = 2m_j c\left(\vec{r}_j - \vec{g}_i\right).$$



(a) (b)

Figure 4.5: Interactive force fields applied to a live molecular dynamics simulation in VR. A) A user applying a force to a carbon atom, highlighted with a white ring, by pulling the trigger on the tracked controller which is rendered in the scene. The interaction is centred at the purple sphere on the tip of the controller, and a graphical representation of the energy in the form of an oscillating wave is drawn between the controller and the atom. B) Comparison of the force that results from the two different interactive potential formulations.

The Gaussian field has the advantage that the maximum force is limited by the Gaussian height, while the spring has no limit. To prevent instability in the molecular system, the maximum force a user can apply is clamped to a maximum value that can be specified by the user and defaults to 20000 kJ/(mol*nm). The difference between the two potentials is shown in Figure 4.5, using the mass of a carbon atom (12 a.m.u) for m_j, a value of 1000 kJ/(mol*a.m.u). for c for the spring potential, and a value of 2000 kJ/(mol*a.m.u) for the Gaussian potential. For the Gaussian potential the force is maximised at a distance of 1 nm from the target, then decays as the user moves their controller further from the atom, while the spring potential maintains a constant force as distance increases. The Gaussian potential has more avenues to tune the strength of the potential, and its decay reduces the chance of accidentally exerting a large force on an atom. On the other hand, the spring potential may be more intuitive as one can simply increase the distance to increase the strength. Determining which interactive potential is better for particular applications remains a question for further study in user tests.

Group Interaction Potential

While interacting with single atoms is sufficient for some uses, such as moving small, inflexible molecules, manipulations of large structures such as protein secondary structures are not possible since applying a large force to an individual atom would alter the structure. To be able to manipulate such structures, it is necessary to apply a force to all the atoms in the structure simultaneously, resulting in a translational motion that preserves the internal structure. Given a set of N atoms that are to be moved as a group, let $\vec{x}_N$ be the centre of mass of the atoms. When an interactive potential is applied to this group, it is calculated as in the single atom case, but instead of a single atomic position, the centre of mass of the group is used as the centre of the interaction, substituting $\vec{x}_N$ for $\vec{r}_j$ and setting m_j to one. This calculation, $\vec{f}_N$, provides the overall direction and scale of the force applied to each atom. This total force is divided amongst the atoms and applied in a mass-weighted fashion via

$$\frac{dV_{ext}}{d\vec{r}_j} = \frac{1}{N} m_j \vec{f}_N.$$

This methodology enables smooth interaction with groups of atoms, enabling complex manipulations.

Velocity Reinitialization Procedure for Interactive Biasing Potentials

Interacting with the atomic system by applying bias potentials enables the motion of the system to be integrated as usual, with the simple addition of the bias. However, the accumulation of biasing forces on the system can have unintended consequences, as the forces are integrated into the velocities of the atoms of the system. This can make systems challenging to control, as the only ways an atom interacted upon will lose the momentum it has built up is either by colliding into something else, through velocity dampening, or friction, from the thermostat, or by the user applying a force in the opposite direction. A reasonable analogy is that it is like pulling a sled on ice by ropes in which the only braking system is the friction felt by the sled. Once the sled picks up speed, the only options are to pull on the rope from the other side to slow it down or to stand by idly as it crashes into something.

Another strategy for interactive molecular simulation is not to integrate the simulation under molecular dynamics, but instead to perform continuous minimization[162]. This strategy works well for small molecular systems and reactions, in which manipulating a single atom and having the system minimise around it is applicable.

Inspired by this strategy, a hybrid method has been developed that uses velocity reinitialization as a way to mitigate the effects of accumulated momentum. Without user manipulations, molecular dynamics takes place as usual. Upon interacting with a single atom, or group of atoms, the molecular dynamics continues to be integrated as usual, except now the interactions forces are also being applied. Once the user stops interacting with the atoms, the atoms involved in the interaction have their velocities randomly drawn from a Maxwell-Boltzmann distribution at a target temperature of αT, where T is the target equilibrium temperature of the thermostat, and $\alpha \in (0, 1]$ is a scale factor chosen by the user, which by default is set to a value of 0.5. Velocities are typically reinitialised to a temperature lower than the target equilibrium temperature to maintain stability. This is similar to the Andersen thermostat, except rather than being applied to atoms at random, the velocity re-initialisation is specifically applied to the atoms involved in an interaction[24]. By reinitialising the velocities, any overall momentum in the atoms in a particular direction is removed. Of course, the system will take time to re-equilibrate, but applying interactive forces already takes the system out of equilibrium, and the benefit of being able to accurately place groups of atoms in a target area outweighs this effect.

4.2.2 Evaluation of the Biasing Potentials

The design and implementation of the group interactive potentials and the velocity re-initialisation procedure were tested by interactively steering the entirety of the small knotted protein MJ0366[194]. This system was chosen because it is flexible but has a specific native conformation, and so it is possible to test whether the group interaction potential preserved structure as designed. The simulation was carried out using the built-in Velocity Verlet integrator at a time step of 1 fs, with the Amber 2010 force field provided by the OpenMM plug-in, with a temperature of 300K maintained with a Berendsen thermostat. The simulation was carried out in a vacuum so that the internal dynamics of the protein and the external potentials were the only effects on the structure of the system. The harmonic interactive potential was used, with a value of 500 kJ/(mol*a.m.u.) for the interactive scaling constant, c. Two trajectories were performed, with interactions beginning after 20 ps of equilibration starting from a minimised state based on PDB entry 2EFV[194]. In the first, group interaction was used to repeatedly move the protein up and down within the periodic boundaries of the system (as shown in Figure 4.6), but without the velocity re-initialisation procedure. In the second, the experiment was repeated, but this time with the velocity re-initialisation procedure ap-

plied after each interaction.

The resulting trajectories were analysed with the analysis package MDTraj[49]. The top panel of Figure 4.6d shows the distance of the centre of mass of the protein from the initial starting condition for each trajectory. The trajectories with and without velocity re-initialisation are shown in orange and blue respectively. The middle panel shows the root mean square deviation (RMSD) from the initial conditions throughout each trajectory, and the bottom panel shows the temperature of the system for each trajectory. Applications of the interactive potentials are indicated by thicker lines in the centre of mass displacement plot. The most important observation to make is that after the protein settles to an equilibrated RMSD, the application of either of the interactive potential schemes does not result in any disruption to the protein structure, indicating that the group interactive potential behaves as desired. There are, however, some differences which illustrate the need for the velocity reinitialization procedure.

Without velocity reinitialization, the motion of the protein is preserved after an interaction, which can be seen by observing that the centre of mass distance continues to increase at the same rate after the interactive potential is applied from simulation time 27 ps to 30 ps. In order to slow the protein down, the interaction potential is aggressively applied again at 31 ps in the opposite direction, resulting in a smooth deceleration before the centre of mass distance decreases as the protein instead moves in the other direction. The user has to repeatedly apply interactive potentials to slow the protein down. By comparison, in the trajectory with velocity re-initialisation the protein loses all concerted momentum when the interactive potential stops being applied, allowing for greater control and more efficient movement. In both schemes, interaction results in the vibrational motion of the protein being dampened out, as the Berendsen thermostat slows the velocities of atoms down in response to the interactive potential. This results in the fluctuations of temperature and RMSD being reduced during an interaction. In the scheme with velocity reinitialization, after the interaction stops the system temperature momentarily drops as all the velocities reinitialise to 150 K, but rapidly equilibrates. The orange line in the RMSD and temperature plots illustrate this. In the system without velocity reinitialization, however, the fluctuations in RMSD and temperature continue to be reduced long after interaction stops, as the translational motion continues to be the most significant contribution to atom velocities.

This experiment demonstrates that a combination of the group interaction and velocity reinitialization methods give much greater control over the molecular system compared to interaction schemes based on applying forces used in previous interactive

molecular dynamics implementations[11, 153].

4.2.3 A Virtual Reality Interface For Interactive Molecular Dynamics

With the algorithms for manipulating complex structures in place, the user experience for actually performing these manipulations in virtual reality has to be developed. Virtual reality presents a new design space for user interfaces and experiences, and so many of the components that are taken foregranted in some of the more commonly used computer interfaces have to be reconsidered. For example, despite some naive attempts by the author, it does not seem viable to bring a QWERTY keyboard with you into the virtual environment, and so many UI crutches such as hot-keys and text-based commands are unavailable. In the years to come, it is likely that new standards for interfacing with VR will emerge that leverage the specific capabilities and limitations of the medium. In what follows, the current design of the VR experience in iMD-VR is presented. As an initial design, it is driven by knowledge of the tasks that users need to be able to perform and follows the design principle of leveraging existing frameworks and interaction methods, enabling rapid development and maximal familiarity to users[195].

Upon initialisation of the simulation, the molecular system is displayed in the centre of the virtual environment at a scale of 1 m in the virtual and physical space corresponding to 1 nm in the simulation. From an implementation standpoint, this initial configuration provides a convenient reference but is arbitrary: the user can change the position and scale of the simulation as required. To move the molecular system, the user can press the 'grip' button on either controller and drag the simulation to any position in the virtual environment. To rotate and adjust the size of the simulation, the user presses both grip buttons on both controllers, and moves their hands in a circular motion to rotate, and draws their controllers together/apart to zoom in/out. This control scheme is a virtual reality analogue to the 'pinch and zoom' control scheme used by many modern touchscreen devices[196] and has been used in other notable VR applications such as Google Tiltbrush[197].

As well as interacting with the simulation, which has been described above, a user needs to be able to easily select groups of atoms in order to restrain them, or to interact with them collectively. Additionally, users need to be able to customise the visualisation of the complex molecular systems. Such operations cannot be achieved with the hard-

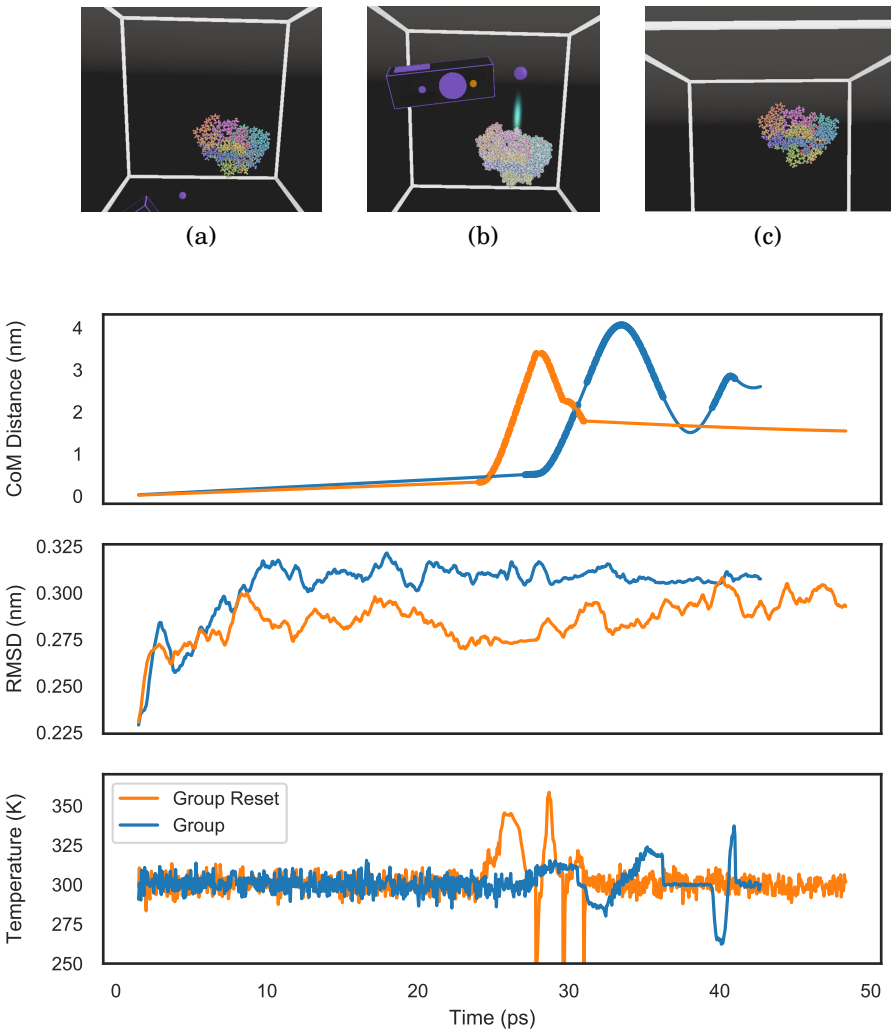


Figure 4.6: Demonstration of the group interaction potential, and the velocity reinitialization procedure in moving the entirety of the protein MJ0366. A) The initial state of the protein at the bottom of the simulation environment, B) interactive potential applied to the protein, with selected atoms highlighted with white rings, C) the protein repositioned to the top of the simulation environment after the interaction. D) The distance of the centre of mass of the protein from its initial conditions, with periods of interaction indicated by thicker lines, the RMSD of over the course of the trajectory and the temperature of the system for two trajectories: one in which the group interactive potential is applied without velocity re-initialisation (in blue) and one where the velocity re-initialisation procedure is applied (in orange).

ware buttons alone, and so in-world UI is required. A commonly used approach in VR is to emit a virtual laser pointer from one of the controllers, which can be used as a cursor to interact with menu panels placed in the virtual environment. The advantages of this method are that it is intuitive for users as traditional 2D UI components such as buttons, toggles and scrollbars behave as expected, and developers can leverage existing 2D UI frameworks to rapidly develop new interfaces. The menus were designed following the Material flat design, ensuring consistency[198].

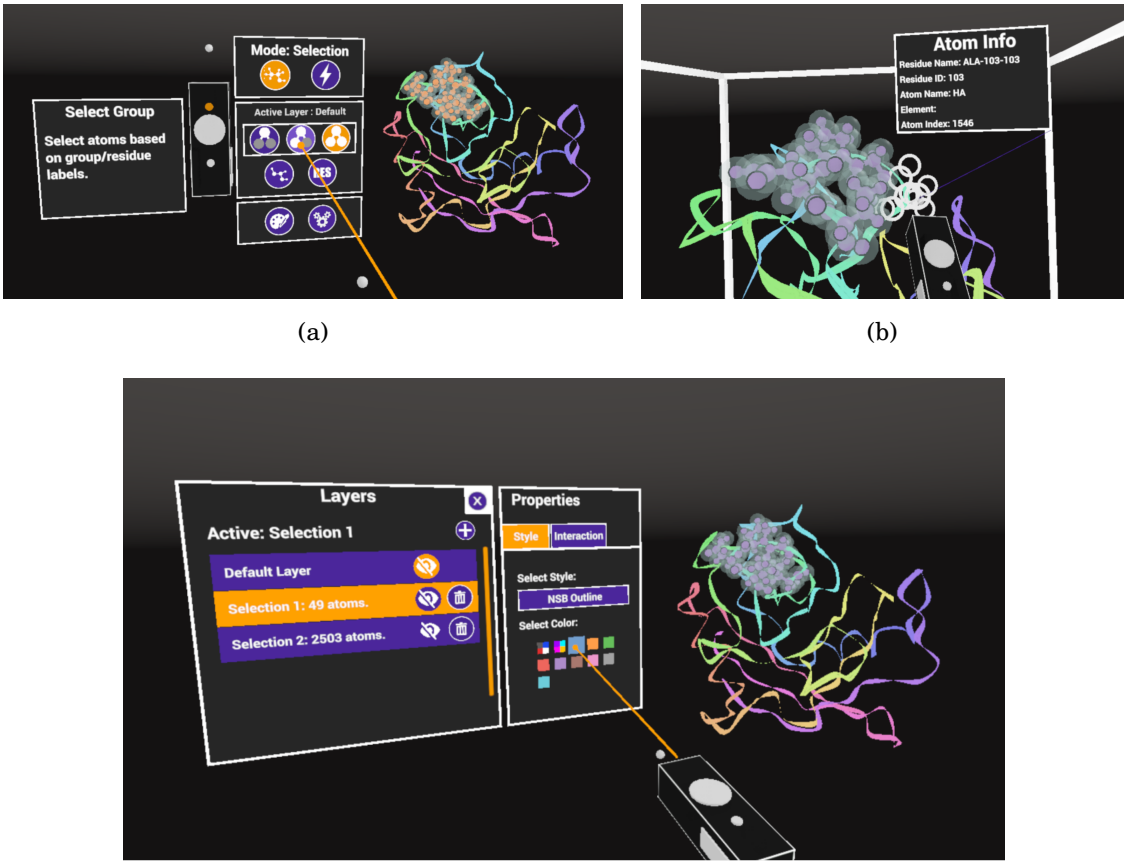
The central UI element available to the user is the menu displayed next to the left controller, depicted in Figure 4.7a. This is a virtual analogue to the artist's palette, allowing the user to quickly change settings on the fly as they interact with the system. Depending on the current state of the simulation, different UI options are displayed. When additional advanced options need to be displayed, these are presented in additional UI panels that appear in front of the user at head height.

The design approach for customisation of rendering and interaction modes that was taken was to enable users to create layers from within the virtual environment, which consist of a selection of atoms, visualisation options, and interaction options. This design will be familiar to many users, as it is similar to the 'representations' in VMD used to define visualisation layers[193], or layers in image editing programs for overlaying and manipulating regions of an image. The purpose is similar; the full molecular system is visually decomposed into a set of overlapping layers to compartmentalise regions of interest. This design gives a wide range of customizability to the user, in which they can create specific visualisations to match their application.

To create layers, a user must select a range of atoms. A virtual reality environment provides a unique implementation opportunity for interactive selection of atoms: a user can literally reach into space and touch the atoms in order to select them, as shown in Figure 4.7b[5]. They can be selected individually, by group information (such as protein residues), or entire molecules or protein chains can be selected. Upon placing the controller near a particular atom, the atom, group or chain are highlighted to indicate that they can be selected, and detailed atom information such as index, residue number, residue name and atom name is displayed. Since many users coming from traditional keyboard and mouse programs typically have some predefined notion of areas of the system that are of interest, it is also possible to select residues in a protein by residue ID lookup.

Upon creation of a layer, a user can choose from many representations and colour

[5]Video available at `https://vimeo.com/306778010`.



(a) (b)

(c)

Figure 4.7: User interface elements for selection and visualisation customisation in the virtual reality application. A) The controller menu, presenting options for selecting atoms. The molecular system (a small knotted protein) is shown on the right, with the user's current selection highlighted with translucent spheres. B) The user is selecting residues in the protein by touching them with the controller. The current selection is highlighted with translucent spheres, and the residue the user is potentially adding to the selection is displayed with white rings. Information about the atom currently being pointed at is shown in a panel positioned near the atoms. C) The user customising the visualisation and interaction settings for the layer highlighted with translucent spheres.

schemes to apply to selection as appropriate, as shown in Figure 4.7c. For interactive simulation, the interactive options that can be applied to a selection are single atom interaction, group interaction, particle restraints, or no interaction. With single atom interaction selection, atoms within a layer respond to interactive forces individually. In group interaction, the entire layer is moved based upon the centre of mass of the atoms in the layer as described above. If particle restraints are applied, each atom in the layer is restrained in its current location with a harmonic potential.

Users' customisations for a particular simulation are stored in a JSON format, which can be transmitted to other users to synchronise visualisation settings. The generality of the layers allows for application specific settings to be created by users in a programmable fashion. For example, users can generate a visualisation scheme from the results of a Waterswap calculation that represents the score of each residue by an appropriate colour[199], or write scripts to port visualisation settings from existing molecular visualisation tools.

These user interfaces enable experts in molecular simulation to perform complex selections for customisation of both visualisation and interaction settings. However, they are merely the first design iteration. In future work, the interfaces will need to be tested under controlled user studies to enable informed iterations that can leverage the unique affordances of virtual reality[200].

Implementation of a Local and Remote Multi-User Virtual Reality Experience

As noted above, the client/server model is inherently multi-user, and so the iMD-VR framework was designed with collaboration in mind. Users can connect to the same simulation from any device, including tablets, desktop computers and of course, other virtual reality setups. In this case, the position of each user's headset and controllers in the simulation's frame of reference is shared via the server with all other users and rendered as simple avatars, resulting in a shared virtual environment in which a molecular simulation can be investigated. Each user can interact with the molecular system independently, and so can coordinate efforts to manipulate the system. This style of multi-user environment is familiar to the video game community in which users have shared a virtual environment through avatars for decades[201].

However, it is also possible to use the dedicated shared space of virtual reality to create an experience, unique to virtual reality, in which users are co-present in the physical and virtual space. By noting that the HTC Vive lighthouses, the infra-red emitters used to provide optical tracking, provide fixed reference points within the space, it is possible

to align the view of all users within the space to create a shared collaborative experience. Let the first lighthouse be labelled A, and the second B. The order is not important but must be consistent across all devices (in the current implementation, the serial numbers of the lighthouses are used to provide this consistency). Let $\vec{r}^A$ and $\vec{r}^B$ be the tracked positions of the lighthouses in the virtual space. The position of the simulation is set to the midpoint between $\vec{r}^A$ and $\vec{r}^B$ and rotated by θ, the angle between the boxes in the xz-plane (Unity3D uses a left-handed coordinate system), which is calculated as

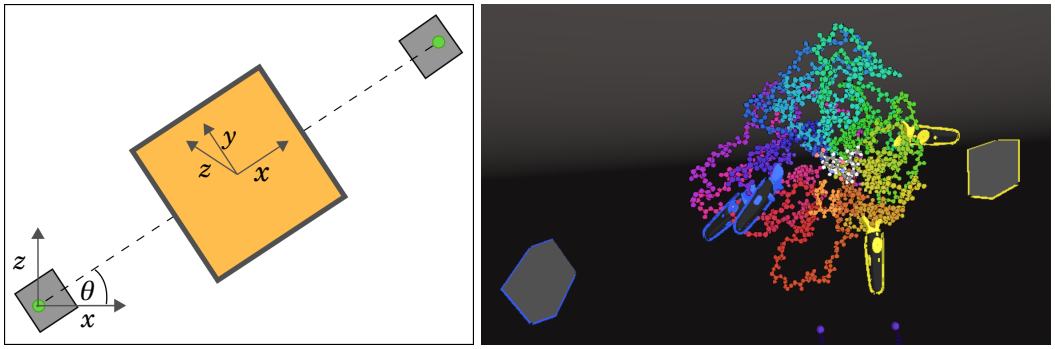
$$\theta = \arctan\left(\frac{\vec{r}^A_z - \vec{r}^B_z}{\vec{r}^A_x - \vec{r}^B_x}\right),$$

to align the simulation to be perpendicular to the lighthouses.

This computation is performed on each VR client within the same virtual space, resulting in the simulation being located physically in the same location for all users. Additionally, the reference frame of the simulation is consistent for all users, meaning that the physical and virtual location of each user and their avatar can be shared between clients and be consistent. While the implementation described here is specific to HTC Vive, a similar approach could be used for other extended reality hardware; the critical requirement is a fixed reference point. The result is a shared virtual experience in which multiple users share the same view of the virtual environment and are located physically where their avatar is. Users can gesture, guide and communicate with each other naturally, and interact with virtual objects in a collaborative manner. For example, Figure 4.8 shows two users coordinating to perform a ligand-binding simulation. This development has a range of implications for the adoption of virtual reality. For research and teaching, the ability to bring others into the virtual environment to inspect a molecular simulation increases its utility as a communication tool.

4.3 Performance of Cloud-Mounted Interactive Simulations

As described above, the client/server model for the interactive simulation framework means that the servers can be hosted remotely on any appropriate infrastructure, be it an academic cluster or a commercial cloud node. The requirements for interactive molecular simulation require high performance in three distinct, asynchronous components: the molecular simulation on the server, the communication to the client, and the VR rendering. Each is crucial: the time to run a time step in the simulation, the lag



(a) (b)

Figure 4.8: Implementation of locally co-located multi-user virtual reality. A) Schematic representation of the method used to align physical and virtual spaces. The SteamVR lighthouses are shown as black outlined boxes. The angle between the lighthouses relative to the Cartesian coordinate system of the virtual scene in the xz plane, indicated by θ, is used to align the simulation. The simulation space, shown as a orange rectangle, is placed at the midpoint between the physical locations of the lighthouses and the internal right-handed coordinate system of the simulation is rotated by θ. B) Depiction of two user avatars in which their virtual location and physical location is consistent for all users. The enzyme H7N9 influenza neuraminidase with bound ligand oseltamivir[202] is manipulated by all users simultaneously, with avatars consisting of outlined cuboids and controllers displayed, positioned by the location of users' headsets and controllers.

time in transmitting the resulting data to the client, and the resulting rendering of the molecular simulation in the updated positions all have an impact on the experience. The development process thus far has been focused on application delivery - an end-to-end prototype that can explore the feasibility of remote interactive simulation in virtual reality, rather than on achieving the highest possible performance in each component. Nevertheless, high performance is a requirement as one scales up to larger systems. In this section, the performance of each of the major components is evaluated.

4.3.1 Molecular Simulation

The target frame rate for a simulation time step is 30 frames per second, or equivalently, 33 ms per step. Below this frame rate, the simulation becomes less responsive to user interactions, and in a virtual reality application this can be uncomfortable.

The compute time for an integration step depends on the complexity of the underlying model and the size of the atomic system. As described, the molecular simulation capabilities of the framework are broad, ranging from classical molecular mechanics to semi-empirical electronic structure. Previous implementations have demonstrated im-

pressive scaling on multi-node systems, simulating up to 1.7 million atoms[159]. However, requiring uninterrupted access to large compute clusters for interactive simulations is a severe limitation. With today's GPU-accelerated molecular dynamics implementations, however, it is now possible to simulate large systems on relatively modest hardware. Figure 4.9 shows the performance of a typical molecular dynamics simulation using the Amber 2010 molecular mechanics force field in OpenMM. The system is a cyclophilin A with increasingly large solvation boxes simulated with the TIP3P water model. A Langevin thermostat was used with a time step of 1 fs, at a temperature of 300 K. A cutoff distance of 2 nm was used for nonbonded interactions, and any bonds involving a hydrogen atom were fixed in length. The simulations were run on an Oracle VM.GPU2.1 virtual machine instance based in Frankfurt with 12 core 2.0 GHz Intel Xeon Platinum 8167M processor, and an NVIDIA Tesla P100. Both the Narupa integrator, calling OpenMM each step for forces, and OpenMM running the PLUMED IMD plug-in described above are shown. The Narupa integrator was run with the mono 5.16 runtime, and the OpenMM plug-in was compiled with the GNU 5.3 C compiler with level 3 optimisations. For comparison, results for simulations running in vanilla OpenMM with no IMD considerations are also shown. The Narupa integrator, unsurprisingly, is the least performant, breaching the 30 FPS target at around 32000 atoms. Simulations running the interactive molecular dynamics protocol in OpenMM, which requires the copying of atomic positions from the GPU every simulation step to transmit to the clients, can run simulations of up to one million atoms before reaching 30 FPS. The IMD protocol results in a decrease in performance of a factor of approximately two when compared to running a vanilla OpenMM simulation.

On the one hand, the results from using OpenMM with IMD enabled are encouraging, indicating that it should be possible to simulate large systems. However, this implementation, which communicates with OpenMM (and other molecular dynamics packages) via PLUMED, has limited capability to manipulate the molecular system. For example, the velocity reinitialization procedure and the ability to reset positions back to a previous time step are not available. While in general one seeks to avoid reimplementation of already optimised molecular dynamics algorithms, unless more aggressive integrations with molecular simulation packages such as OpenMM are implemented, the Narupa integrator provides some crucial advantages and thus should be optimised.

Figure 4.9 shows a breakdown of the method occupancy in the Narupa engine. This breakdown indicates that there are many inefficiencies in the current implementation.

The most significant component is the SETTLE constraint solver, which is currently a very basic implementation with no parallelisation or optimisation. The OpenMM forces calculation is the next largest component. A significant portion of this is in copying data between Narupa managed memory and the OpenMM unmanaged memory, which includes significant memory allocation, as indicated by the breakdown of the force field computation in the right-hand panel of the plot. Setting the positions, reading back the forces, and accumulating them back from OpenMM all take a significant amount of time. This is due to the use of an automated procedure, SWIG, to create a wrapper around the OpenMM library. This approach was taken in the interest of maintainability and ease-of-use but has introduced significant overheads. This an implementation issue that can be resolved, and so it should be possible to achieve scaling closer to that of OpenMM with the PLUMED plug-in.

The other major bottleneck is in propagating the molecular positions and ensuring the molecules are wrapped into the periodic box, following the minimum image convention. The overhead of this procedure could be reduced by introducing parallelism.

From this profiling, it is clear that there are some obvious and straightforward optimisations that can be made which will significantly increase the size of simulations that can be run with the Narupa integrator. Additionally, the current implementation is pure C#, which enables good interoperability with Unity3D and cross-platform support. However, as a just-in-time compiled language, it can be slower than statically compiled languages such as C/C++, and thus it may be beneficial to reimplement some of the core routines in such languages. This would also allow for interoperability with languages more commonly used in scientific simulation such as Python.

Despite the implementation issues, the molecular dynamics compute performance is encouraging, as it indicates that simulations of a scale of interest to biochemists can be simulated at speeds conducive to interactive modelling.

4.3.2 Network Communication

A crucial component when manipulating a remote simulation is the lag time. In the context of an interactive simulation, the lag time is defined as the round trip time: the time it takes to communicate with the server and for it to respond. This is the relevant definition as this is the time it takes for the simulation to respond to user input and for the user to observe the response. As the ping time increases, the lag between selecting molecules to interact with and the force being applied to them increases, with it eventually becoming very difficult to control. With servers in Frankfurt, Germany; Ashburn,

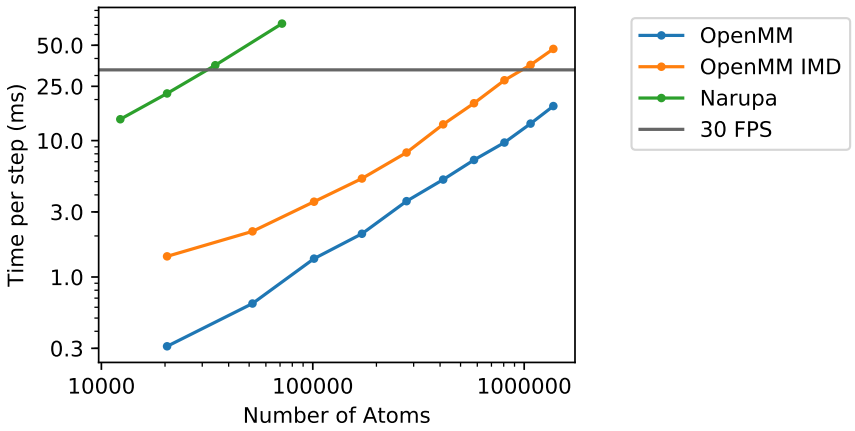


Figure 4.9: Average time to perform an integration step with increasing number of atoms for a classical MD simulation running on an Oracle VM.GPU2.1 virtual machine instance based in Frankfurt with 12 core 2.0 GHz Intel Xeon Platinum 8167M processor, and an NVIDIA Tesla P100. The green line indicates the performance of the Narupa integrator, using the OpenMM CUDA force field. The blue line indicates the performance of unaltered OpenMM, while the orange line indicates the performance of OpenMM with interactive MD transmissions occurring every frame via PLUMED. The grey line indicates the 30 frames per second target.

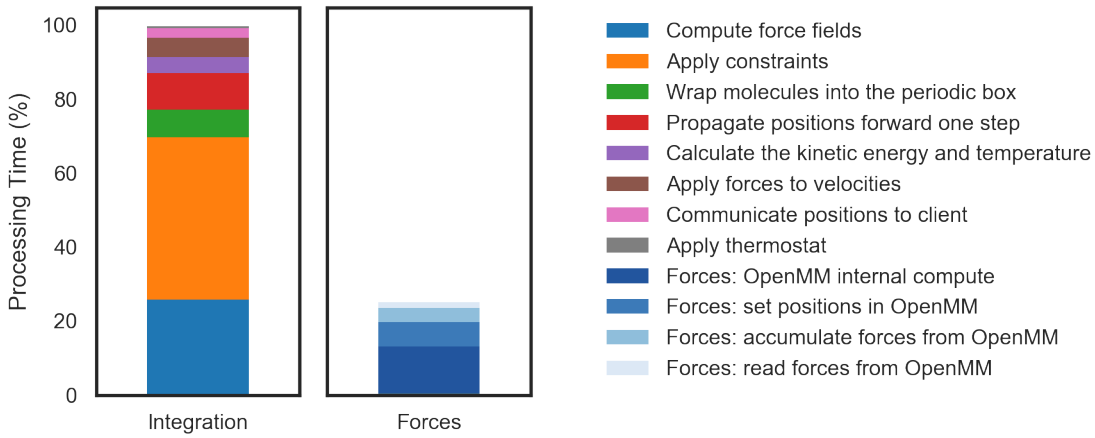


Figure 4.10: Detailed profiling of molecular dynamics in iMD-VR. The left-hand panel shows the breakdown of time spent in various operations per integration step, while the right hand-panel shows a detailed breakdown of the force calculation with OpenMM.

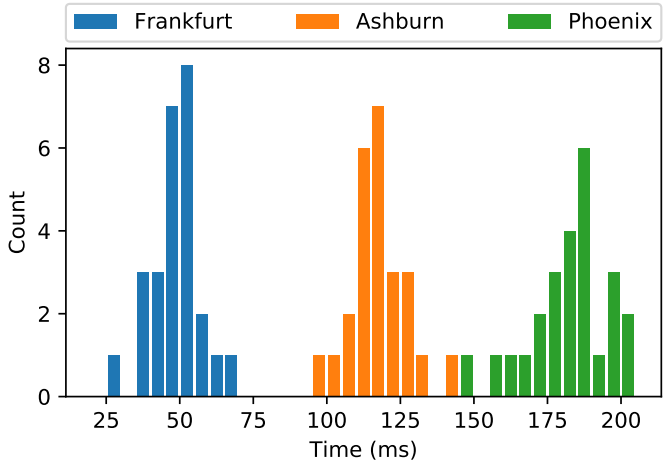


Figure 4.11: Distribution of round-trip times from University of Bristol, U.K., to a server running on each of the three cloud centre locations: Frankfurt, Germany, Ashburn, Virginia, U.S.A, and Phoenix, Arizona, U.S.A.

Virginia, U.S.A; and Phoenix, Arizona, U.S.A; it becomes possible to provide a quantitative evaluation of how increasing distance from servers over commercial network infrastructure affects performance. Figure 4.11 shows the distribution of ping times from the University of Bristol, U.K., to each of the three cloud centres running the simulations available in the application described in Chapter 5. The ping time was measured from within the application, and so is an accurate representation including any software stack effects. Unsurprisingly, the average ping time increases with distance from the server, as does the variance of ping times. The ping times to Frankfurt have a mean of 47 ms, with a standard deviation of 8 ms, which results in a smooth user experience. With mean lag times of over 117 ms for the servers in the USA, manipulation of molecules becomes challenging. Therefore, for interactive molecular simulation, the location of the remote servers and the quality of the infrastructure is an important consideration.

To test the effect of increasing system size on the transmission of data, a variety of simulations were run on the GPU node in Frankfurt described above and connected to from Didcot, U.K, over a WiFi connection. The effect of increasing numbers of atoms was measured from an application-centric view, by measuring the rate at which the client was receiving frames from the server. This measures the performance of the entire transport stack, including the implementation of the WebSockets protocol as well as the data transfer over the wire. These included TIP3P water boxes of increasing size and ideal gas simulations of increasing numbers of atoms. Additionally, a server which

directly communicated the initial set of atom positions at a rate of 30 frames per second and performed no integration at all was tested, to investigate the effect of the computational workload.

Strip plots of the frame receive time as a function of the number of atoms are shown in the left-hand panel of Figure 4.12. The distributions are very narrow with long tails, as shown in the right-hand panel which plots the percentiles of the distribution of receive rates for increasing number of atoms, showing the proportion of observations that were received within a given time. The flat shape of each of these distributions up to the 95th percentile indicates that the majority of frames are received at a consistent rate, with a long tail of outliers that take longer to arrive. For small simulations of up to 12000 atoms, the receive rate is mostly at the target rate of 33 ms, as shown by the grey horizontal line. As the simulation size increases, the average receive rate increases, as does the variance, with more outliers in which the time between frames can be momentarily on the order of hundreds of milliseconds. Beyond 12000 atoms, the average frame rate trends upwards beyond 33 ms at which point noticeable lag becomes apparent in the application. A simulation of 12000 atoms corresponds to the transmission of the array of 36000 half-precision floating point numbers 30 times per second, or equivalently a throughput of 2.16 MB/s.

The observed receive rates are encouraging, considering the distance to the server, showing that the network layer is reliable as it delivers frames at a consistent rate. It is likely to be possible to optimise the transport implementation, but the result that 12000 atoms can be comfortably transmitted in real-time is encouraging, as it means that systems of a meaningful size can be hosted remotely. According to an analysis by Karlin and Brocchieri, the median length of a representative dataset of eukaryotic proteins is 375 amino acids[203]. The mean number of atoms across all amino acids is 19.2[204], and so an estimate for a typical number of atoms is around 7200 atoms. This back-of-the-envelope calculation suggests there are many systems of interest to biochemists that are accessible to the current implementation if one does not transmit the positions of any explicit solvent. Assuming a user wishes to perform larger simulations (which is inevitable), such an atom-centric view of data transmission is not necessarily the most relevant. While the simulations are performed at an atomistic level, analysis and visualisation are typically at a coarse-grained level, with a few areas of detail. Instead, it is better to consider the limit of 12000 atoms as a guide of 12000 positions, or more generally 72 kilobytes per simulation frame. Within this limit, one seeks to optimise the transmission of data that is of maximal utility to the user. This idea is amplified when

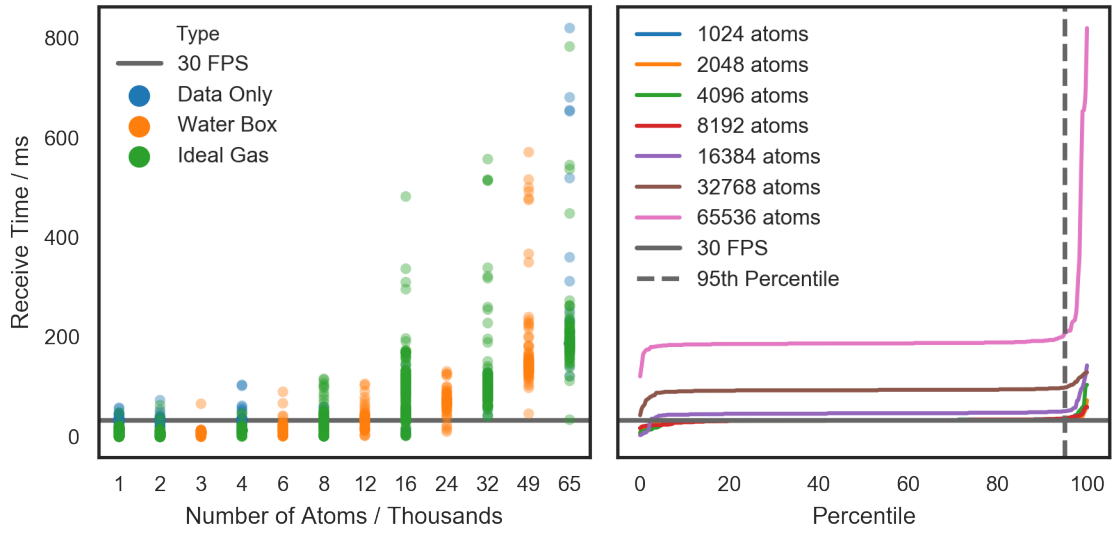


Figure 4.12: Effect of increasing simulation size on the client receive time for cloud based-simulations. The left-hand panel shows strip plots of the receive time for a variety of workloads with an increasing number of atoms. Ideal gas simulations are shown in green, simulations of water are shown in orange, and sending data with no simulation running is shown in blue. The right-hand panel shows percentile plots of the receive time for increasing number of atoms in an ideal gas simulation.

considering the requirements for real-time rendering.

4.3.3 Real-time Molecular Trajectory Visualisation in Virtual Reality

In a VR application, the target frame rate is 90 frames per second for each eye or 180 frames per second in total[205], which is substantially higher than traditional video game or other desktop application refresh rates (typically 30 frames per second on video game consoles, and up to 60 frames per second on personal computers)[206]. This high frame rate is required to ensure user comfort, as lower frame rates are more likely to result in discomforts such as headaches and motion sickness. In molecular visualisation, there are many different ways to represent molecules, all of which have different performance characteristics[153]. An additional consideration for interactive molecular simulation is the requirement for rendering to take place in real-time. Unlike other molecular viewers, where a pre-recorded trajectory is processed and then rendered, an interactive simulation is a stream of data that is continuously updating. Thus the atom position streams have to be processed as they are received and then passed to the ren-

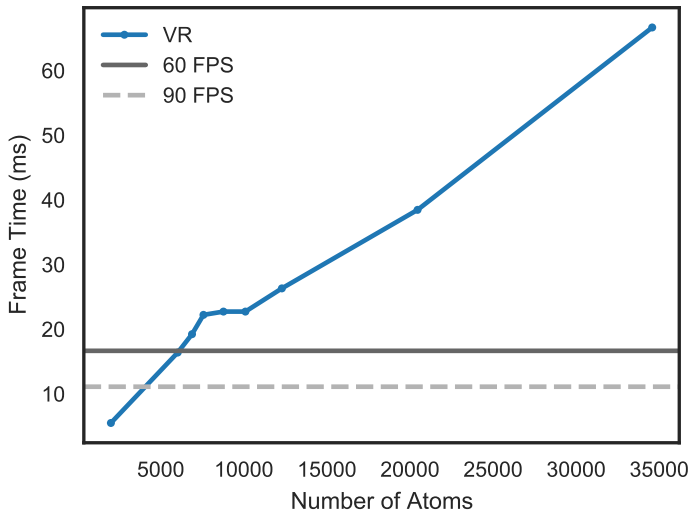


Figure 4.13: Average time to process a visualisation frame in the virtual reality client with increasing number of atoms. The blue line shows the performance of the virtual reality client, while the dashed and solid grey lines indicate the 90 and 60 frames per second performance targets respectively.

dering pipeline.

As noted above, the molecular renderers were developed by developers at Interactive Scientific, and are not the focus of this thesis. Nevertheless, understanding their current performance characteristics is essential for future development.

Figure 4.13 shows the measured frames rendered per second of the virtual reality client with increasing number of atoms, using the default ball-and-stick model which, as shown in Figure 4.15, consists of medium quality hemispheres (icosahedron with 40 vertices), mesh bonds, and outlines rendered using billboards, all rendered using Unity3D particle systems. The simulations are the same as those used for profiling the molecular dynamics (cyclophilin A in increasingly large solvents of explicit water). The VR headset was placed in a fixed position during measurements. The benchmarking was performed on an Alienware15 R3 gaming laptop with a 2.6GHz quad-core Intel Core i7-6700HQ and an NVIDIA 1070 GTX dedicated VR graphics card, with an HTC Vive Pro headset with a resolution of 2880 by 1600 pixels. This laptop meets the requirements for running virtual reality hardware but is not an exceedingly high-end machine.

Up to around 4000 atoms, the virtual reality client achieves the target 90 FPS per eye, but drops to 60 FPS at 6000 atoms and then continues to degrade beyond that. Profiling in the Unity3D editor was undertaken for a simulation of 30000 atoms, to determine which processes need optimisation. The processing largely takes place in two

threads, the main CPU thread and the rendering thread. The left-hand panel of Figure 4.14 shows the contribution to the total time per frame, shown on a logarithmic scale, for the largest CPU processes. Processing the updated positions of the bonds, spheres and the size of outlines is a significant bottleneck, taking up to 24 ms per frame, while the components of the rendering performed on the CPU are reasonable, at 0.93ms.

The right-hand panel shows the rendering time, on a logarithmic scale, for the available base renderers. The point and lines renderers display 2D atoms and bonds respectively using sprites, the VDW renderer displays spheres (of adjustable quality), and the bond renderer displays cylinders (of adjustable quality). As described above, the outline renderer combines low-quality spheres and bond renders with the point and line renderers to display outlines. The point and line renderers are very efficient, with a rendering time of just over 1 ms for 30000 atoms, while the mesh based renderers introduce significant overheads. The quality of the spheres has a tremendous impact on performance, with the highest quality spheres, with 256 vertices each, taking over 200 ms to render. Combining a lower quality sphere of 40 vertices with the points renderer to produce an outline gives a much better performance of 26 ms per step, and the outlines effectively obfuscate the lower quality sphere.

It is clear from the profiling that there is a lot of room for improvement in the rendering performance. Working with a game engine such as Unity3D provides an excellent platform for rapid development of user interfaces and portability to many platforms, but it can be challenging to balance this ease-of-use with performance. The profiling suggests that the processing of atomic data on the CPU needs optimisation and that billboard representations should be used for the display of large systems at atomic resolution. Processing on the CPU currently takes place on a single thread, and so an obvious optimisation is to introduce multithreading. However, efficiently constructing a multithreaded pool in a manner that interacts well with Unity3D and does not introduce CPU pressure from too many threads is not trivial. A recent update to Unity3D introduces a job system that will make this easier[207]. It is not currently compatible with the particle system pipeline used to batch the atom rendering, but this is expected to change in an upcoming update. It should then be possible to optimise the processing of atomic data.

Alternatively, compute shaders have now become available in Unity3D, which allow for general-purpose GPU code to be written in a manner similar to that of CUDA or OpenCL[208]. With this, it ought to be possible to perform the atom processing on the GPU and pass it directly to shaders for rendering. Advances in real-time ray-tracing

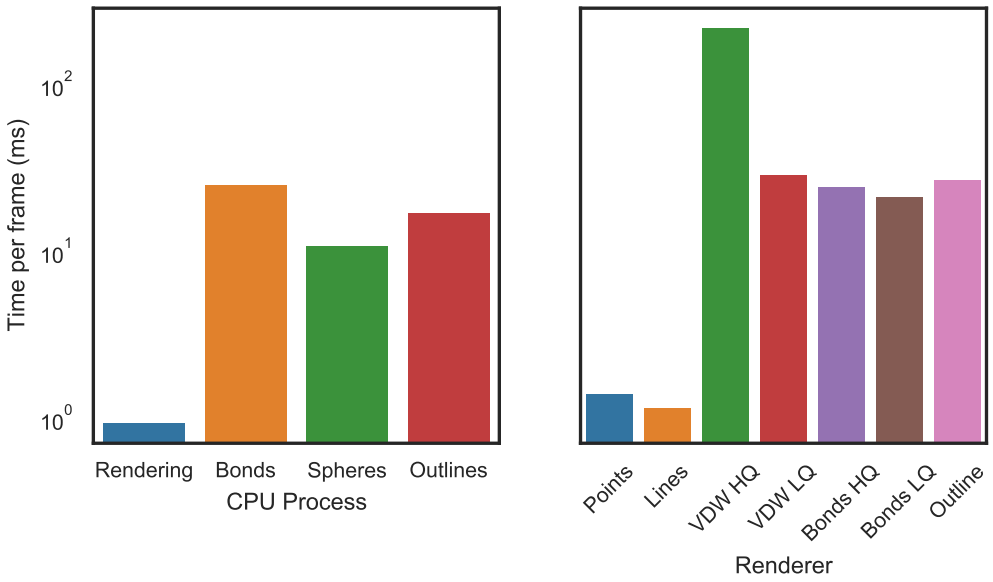


Figure 4.14: Profiling of the VR application running a simulation of Cyclophilin A in an explicit solvent, totalling 34577 atoms. The left-hand panel shows the time spent per frame on various CPU processes, including rendering, processing bond representations, atom positions and the position of outlines. The right-hand panel shows the time per frame for the available renderers. The Points renderer is a billboard renderer of simple circles, the Lines renderer is a billboard renderer of simple lines, the VDW HQ/LQ renderers are spherical meshes with 256 and 40 vertices respectively, the Bonds HQ / LQ renderers are prism meshes with 9 and 5 sides respectively, and Outline renderer combines the VDW LQ renderer with the Bonds HQ renderer, and Points and Lines renderers to provide an outline.

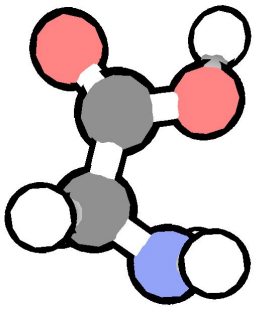


Figure 4.15: Example of the default outline renderer, showing a glycine molecule. Rendered using the VDW LQ renderer for the sphere, the Bonds HQ renderer for the bonds, and a Points and Lines renderer for the outlines.

on GPUs may make it possible to produce the high-quality, high-performance rendering in virtual reality that is already familiar to users of other molecular visualisation programs[209].

While there is no doubt the rendering could be improved, as the current implementation has not been heavily optimised, there is a question of the value of rendering thousands of particles explicitly.

Given that it is possible to run simulations that are much larger than can both be transmitted and rendered at atomistic detail, there is a lot of scope for optimisations in the choice of what data is transmitted to the user and how it is presented. This is an area at the cross-section between high-performance computing and user experience design, as one seeks to intelligently identify what needs to be represented at any given time. For example, it is common for users to choose not to render solvent in virtual reality, and not even to transmit them to the client application. Furthermore, for biomolecular applications, a user is typically only interested in the secondary structure of most of the simulation, with a few crucial regions rendered in more detail. Rendering secondary structure is typically substantially less expensive than a full atomistic representation, as a simple mesh can be generated that only requires the position of the atoms in the backbone of the protein. A common approach in computer graphics is to use 'level-of-detail' (LOD) method in which objects are rendered at lower quality as the distance from the camera increases[210]. This is certainly applicable to molecular rendering. In future research, it would be desirable to optimise the dataset that is transmitted to the client based on what the user is looking at and their current visualisation options. The modular and extensible streams architecture of the framework will enable rapid development of transmission protocols for specific applications.

4.4 Conclusions

In this chapter, the concept of interactive molecular dynamics as a means to sample molecular conformations was reviewed, with the limitations of previous attempts discussed. A framework for multi-user interactive molecular simulation was described which can be deployed on remote computer infrastructure, and novel algorithms and design considerations for a virtual reality environment for interactive molecular dynamics were presented. The modular, extensible framework provides a complete set of molecular dynamics algorithms, communication protocols, molecular visualisation and interaction methods appropriate for manipulation of complex molecular systems. The

scaling performance of the major components of the system was evaluated, and while there is plenty of room for optimisations, the current implementation was found to be sufficient for proof-of-principle applications. In order to scale up to larger systems, optimisations in the communication with molecular dynamics plug-ins are required, new rendering technologies will need to be explored, and intelligent filtering of the data that is transmitted and how it is represented to the user will need to be considered.

Interactive molecular dynamics and virtual reality provides an intuitive way to identify pathways and conformations rapidly, and the framework has been developed explicitly as a method for finding initial pathways and configurations that rare event algorithms will be able to leverage. The following chapter utilises the interactive molecular dynamics framework described in a user study to evaluate the utility of virtual reality in performing various molecular tasks. In subsequent chapters, the framework is evaluated in a number of preliminary applications for finding pathways and conformations in complex molecular systems.

While not the focus of this thesis, the platform has broad potential beyond the goal of accelerating molecular simulation. It can also be used as an analysis tool for existing molecular trajectories and structures. The multi-user design allows for collaboration in this analysis, as well as opportunities for it to be used in the teaching of molecular science. In recent work, it has been used in undergraduate teaching to complement a molecular dynamics teaching lab. A cohort of students were given molecular tasks to complete in VR, such as a rearrangement of chorismate in vacuum, and binding of chorismate with chorismate mutase. A student survey indicated that not only did most students find the VR component more engaging but that it also improved their perceived educational outcomes and their interest in continuing on in the field of computational sciences. A publication of this work is in preparation.

CHAPTER 5

ACCELERATION OF MOLECULAR TASKS WITH INTERACTIVE MOLECULAR DYNAMICS

A reasonable question, given the repeated rise and fall of consumer interest in VR, is whether VR interfaces, while undoubtedly providing a more immersive experience compared to traditional devices, actually provide any measurable benefit in the undertaking of tasks within the virtual environment. For some tasks dependent on 3D reasoning, such as medical surgery, there is an existing body of evidence to support the notion of VR being an effective tool for training surgeons to complete surgeries faster with a lower error rate[178]. It stands to reason that manipulating 3D molecular structures may also benefit from the use of VR, but it is crucial to evaluate this claim if it is to be used to accelerate rare events in molecular systems.

This chapter, adapted from Ref [149], presents a user study that was undertaken to test this claim. The study exploited the fact that the interactive molecular dynamics framework described in the previous chapter can be easily configured to run on a variety of devices with different input schemes. These include the aforementioned virtual reality interface using the HTC Vive, a desktop set-up using a 2D display with mouse and keyboard for input, and an implementation for smartphones and tablets, which are referred to as the VR, mouse and tablet set-ups respectively for the remainder of the chapter. Three tasks for users to carry out were designed, with increasing complexity intended to challenge 3D reasoning. The tasks were threading a methane molecule through a carbon nanotube, reversing the screw-sense of a helicene molecule, and tying

a knot in a small 17-ALA polypeptide. A total of 32 users were tasked with performing each of these tasks on the different platforms within a specified time and surveyed for qualitative feedback on the experience of using each platform.

My contributions to this study included the development of the software and algorithms discussed in the previous chapter, the configuration and implementation of the simulations used in the study, and development, in collaboration with developers at Interactive Scientific Ltd, of the specific applications developed for the study that were made available for download[1]. PhD student Helen Deeks and undergraduate student Edward Dawn performed the user study, gathered user feedback and performed interviews, and Helen Deeks performed the analysis of the resulting data.

5.1 Methods

5.1.1 Outline of User Tasks

The following molecular tasks were designed for users to carry out, and are illustrated in Figure 5.1.

- Buckminsterfullerene task: A simulation of two buckminsterfullerene molecules was run in the iMD-VR framework. Participants were instructed to spend as long as they liked familiarising themselves with the control schemes, e.g. moving the molecules and reorienting themselves and the simulation.

- Nanotube/methane task: A simulation of a short open-ended carbon nanotube and a methane molecule was run in the iMD-VR framework. Participants were instructed to use the interactive potentials to guide the methane through one end of the nanotube and out the other. The task had a time limit of 180 seconds, and was considered completed when the participant successfully pulled the methane out of the nanotube from the opposite side to which it entered.

- Helicene task: A simulation of a single 12-helicene molecule was run in the iMD-VR framework. Participants were instructed to use the interactive potentials to reverse the screw-sense of the molecule. The task had a time limit of 210 seconds.

- Protein knot tying task: A simulation of a 17-alanine peptide was run in the iMD-VR framework. Participants were instructed to tie the protein into a simple trefoil

[1] Available at https://isci.itch.io/nsb-imd.

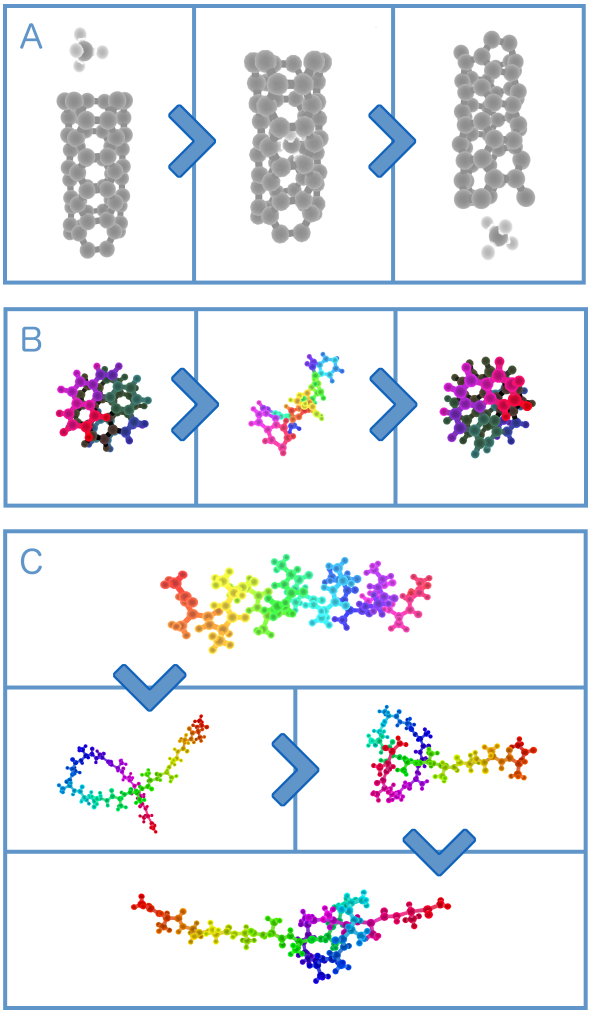


Figure 5.1: Interactive molecular simulation tasks used as application tests: (A) threading a methane molecular through a nanotube; (B) changing the screw-sense of a helicene molecule and (C) tying a knot in a polypeptide. Colours selected in this figure are chosen for the sake of clarity in representing 3D objects on 2D media.

knot. Due to the complexity of the task, participants were shown a video of the task being completed prior to participating. The task had a time limit of 180 seconds.

These tasks were designed to require a range of 3D reasoning and to be small, accessible analogues to important applications across nanomaterials and molecular biology in which IMD may prove useful. For example, transport of materials through nanotubes is an active area of research[211] and is similar in spirit to the transport of molecules through pores and channels in cell membranes. While small peptides are not thought to tie themselves into knots spontaneously, a significant number of proteins are now

known to exhibit knots in their native structure[212]. Indeed, to the author's knowledge, a trefoil knot in a small peptide has never been demonstrated previously, providing an indication of the possibilities of interactive molecular dynamics in the exploration of conformational space. The stability of such conformations is an area of future study.

5.1.2 Simulation Methods

The simulations used for performing the tasks were developed for use with the interactive molecular dynamics framework described above. A specific client application was developed for each platform. The VR application was a simplified version to that described in Chapter 4.2, with all advanced features and customisations unavailable to the user. The user could rotate, resize and move the simulation using hardware buttons and gestures on the VR controllers, and could interact with the simulation by reaching into the simulation and pulling the trigger, as described above. In the desktop and tablet versions of the application, the user cannot simply walk around the molecule to adjust their viewpoint. Instead, the simulation was displayed in the centre of the field of view of the application, and, in a strategy used in other molecular visualisation applications[193], the user could rotate the camera around the centre of the simulation and zoom in and out. On the desktop version, this was achieved by clicking and dragging to orbit the camera, and scrolling of the mouse-wheel to zoom in and out. On the tablet version, a single finger press and dragging motion orbited the camera, while a two-finger 'pinch and zoom' motion was used to zoom in and out.

A different method of interaction with the simulation on the tablet and desktop versions of the app was required, as the user cannot specify precisely where in the 3D space they wish to apply the interactive potential. Instead, upon clicking the mouse or tapping the screen, a ray was cast into the simulation from the point on the screen at which the user clicked. The first atom to intersect with this ray was then selected as the atom to be interacted with, and the interactive potential field was placed at this point. While holding the mouse button down or sustaining a finger on the screen, the user could then guide the atom by moving the potential. While interacting with an atom, the potential was moved in the 2D plane that intersected the atom and was parallel to the camera orientation, as shown in Figure 5.2. On all platforms and all tasks, a value of 2000 kJ/(mol*a.m.u.) was used for the scaling constant c, balancing responsiveness with dynamic stability, the Gaussian potential was used, and only single atoms could be interacted with. To indicate which atoms had been selected, all other atoms darken in colour except for the atom selected (see Figure 5.2). This was found to be a particularly

effective form of visual feedback on the tablet version of the app, where the user is often trying to manipulate an atom that is hidden under their finger, and so by darkening all other atoms, it was clear that the correct atom had been selected.

The implementation of the molecular simulation for each task leveraged the modular implementation of the IMD framework, using different forcefields as appropriate. For the buckminsterfullerene, carbon nanotube and helicene tasks the MM3 forcefield was used, while for the knot tying task the GPU-accelerated Amber99SB-ILDN forcefield provided within the OpenMM molecular dynamics package was used. The forces were integrated with a Velocity Verlet integrator, and a Berendsen thermostat was used to maintain a target temperature of 200K for all tasks. This low temperature was used to ensure stability and to make it easier for users to manipulate. For the nanotube and helicene tasks, a time step of 1 fs was used, while the protein knot task used a time step of 2 fs, and hydrogen bonds were held at a fixed length with holonomic constraints.

The simulations used in the user studies were hosted on separate machines within a local area network. The simulations were hosted on local machines because at the time the study was undertaken the cloud-mounted simulation architecture was not yet available. One machine was used for each task, in order to avoid circumstances where latency could arise from an excessive computational load on the machine. The three machines that were used as simulation servers during the user studies included the following: (1) a mid-range gaming desktop with a 3.5GHz quad-core Intel i5-6600K processor and a Nvidia GTX 970 graphics card; (2) a high-end Alienware13 R3 gaming laptop with a 2.6GHz quad-core Intel Core i7-6700HQ processor and an Nvidia 1060 dedicated VR graphics card; and (3) a high-end Alienware15 R3 gaming laptop with a 2.6GHz quad-core Intel Core i7-6700HQ and an Nvidia 1070 GTX dedicated VR graphics card. The mouse tests described in the first user study (run at the lab in the Bristol chemistry department) were carried out using machine (1) along with an external USB mouse and a 21” external monitor. The mouse tests described below in the third user study (run at the CCP-Biosim 2017 conference) were carried out using the screen on the Alienware15 along with an external USB mouse. During all user studies, the simulations integrated the simulations at a rate of 30 steps per second on all platforms. This ensured that the latency across all test platforms gave an equally fluid user experience. For the touchscreen version of the tasks, we used a Samsung Galaxy S3 tablet, connected to the simulation over an 802.11ac Dual Band Gigabit Wi-Fi connection.

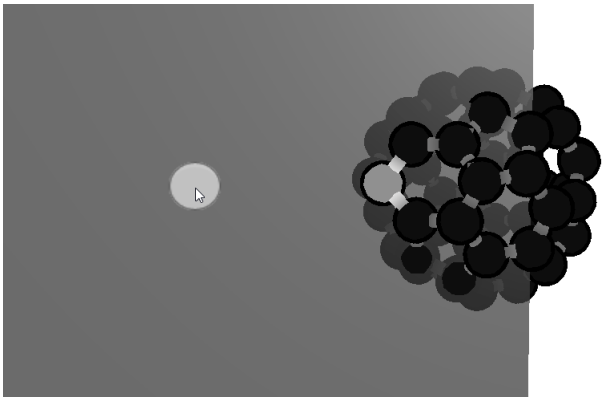


Figure 5.2: Screenshot of the interactive potential interface on the tablet and desktop versions of the IMD application. The light-coloured atom on the buckminsterfullerene molecule is the selected atom, and the translucent grey circle is the position of the interactive potential. The plane of interaction between the selected atom and the interactive potential, not usually displayed in the application, is shown here in translucent grey to illustrate the degrees of freedom available to the user.

5.1.3 User Study Design

The experiment took place in the form of three separate user studies which broadly share the same methodology but with a different focus and are outlined below.

In the first user study, a total of 32 participants were tasked with the carbon nanotube and helicene tasks described above. To mitigate any learning or fatigue effect, participants started the study on one of the three platforms (mouse, tablet and VR) at random, and then rotated through the remaining platforms such that all participants undertook the tasks on all three platforms. Participants were recruited from staff and students at the University of Bristol by email, enticed by a £10 Amazon gift voucher for their time.

Before beginning the timed tasks on any platform, participants were introduced to the buckminsterfullerene simulation to familiarise themselves with the particular experience of molecular interaction on the platform. Once a sufficient familiarity had been gained, the participant then moved onto the nanotube/methane task, in which they were timed. The study facilitators moved the participant onto the next task, the helicene task, immediately after they had completed task or after the allotted time expired if the participant was unable to complete the task. This process was repeated for each task on each platform until all participants had attempted all tasks on all platforms. Upon finishing all the platforms, participants were then given a short questionnaire to fill out to provide some insight into the participant demographics, their familiarity with each

platform, and feedback on the experience. The participant's self-reported familiarity with VR and tablet touchscreen interfaces were recorded on a Likert scale, where 1 represents no experience, and 5 represents a great deal of experience. Given the age and education level of the participants, familiarity with mouse and keyboard interfaces was assumed to be high. To probe into what features of the VR interface, in particular, were deemed important or useful for achieving tasks, participants were asked to rate on a Likert scale from 1 to 5 the importance of some previously identified aspects of the interface. These included the importance of depth perception, the importance of being able to physically move around the simulation, and the importance of being able to interact with two atoms simultaneously, where one indicates not important at all and five indicates very important. Finally, general comments on the experience were gathered by asking the participants to note any benefits or difficulties in using each platform.

A second study was performed in which 12 participants were tasked with performing the knot-tying task on each platform, with an emphasis on gaining qualitative feedback from participants on using the different platforms by interviewing them after attempting all three tasks. Feedback gathered included ascertaining which platform (if any) was preferred, and why, as well as general feedback on the user experience. Again, the participants were randomly assigned to an initial platform and used the buckminsterfullerene simulation as a tutorial.

A third study was undertaken in which the methodology of the second study was repeated with 32 participants undertaking the knot-tying task on each platform. In this study, there was no follow-up interview as the goal was to gather more data on task accomplishment rates in the knot tying task. Participants were recruited during the 5^{th} annual UK CCPBioSim conference, held at the University of Southampton.

5.2 Results and Discussion

5.2.1 Task Accomplishment

Table 5.1 and Figure 5.3 shows the accomplishment rate and times of the participants for each task on each platform. The accomplishment rate provides insight into what proportion of the cohort completed each task on each platform, while the accomplishment time indicates if the task could be completed faster on any particular platform. The error bars shown in the accomplishment rate plot were calculated using Poisson

statistics[2]. The error bars indicate that rates are statistically distinguishable if the error bars do not overlap, and so provide a reasonable way of determining if a platform provided any benefit.

Participants performed reasonably well on tasks when using the VR platform, with 97% able to accomplish the nanotube task, 47% able to change the helicene screw sense and 72% able to achieve the knot-tying task. This indicates that the tasks varied in difficulty, as intended, and were challenging enough to require competence with the interface. The mouse and tablet interfaces are comparable, indicating that users did not find either to give any particular advantage.

For the nanotube and knot tying tasks, the virtual reality interface provides a clear, statistically significant benefit. The average accomplishment rate of the nanotube task on the VR platform was nearly twice that of the mouse platform, and for the knot tying task the rate was an order of magnitude greater. Threading the methane through the nanotube requires precise control in 3D space, as both the nanotube and methane are freely rotating and moving. On the VR platform, one can use one controller to position the nanotube, and the depth perception provided by the HMD to accurately position the methane to thread it through. Similarly, the knot task requires a complex sequence of positioning, reorientation and motions in multiple directions in 3D space, which is extremely challenging on the 2D interfaces[3].

The helicene task stands out in that there is no significant difference in accomplishment rates between the platforms. Upon considering the manipulations required to achieve the task, it becomes clear that this result is reasonable. The 'trick' to this task is to realise that it can be achieved by selecting an end of the molecule and rotating it counter-clockwise in the plane orthogonal to the lengthwise axis of the helicene molecule. This reverses the helicity of the end grabbed, which with continued application propagates down the rest of the molecule. It is effectively a 2D motion that can be achieved with ease on all platforms if one discovers it.

Additionally, for the nanotube task, assuming a participant completed a task on a given platform, they were more likely to complete the task more quickly with the VR platform than the other platforms. A Welch's t-test[213] was performed with the python package Scipy[214] between each pair of platforms for each task, testing the hypothesis that the mean completion time was lower on a given task with one platform compared

[2]Error is calculated as $\pm 0.5 + \sqrt{n + 0.25}$, where n is the number of observations.

[3]The author, despite high familiarity with all platforms, has never managed to achieve this task on the touchscreen platform.

Platform	Task					
	Nanotube		Helicene		Knot	
	Mean (s)	Rate	Mean (s)	Rate	Mean(s)	Rate
Mouse	110 ± 44	16	124 ± 56	14	120 ± 30	2
Touchscreen	106 ± 59	13	107 ± 54	14	n/a	0
VR	64 ± 42	31	107 ± 62	15	86 ± 48	23

Table 5.1: The task accomplishment rates and mean completion times by participants in the study. The average completion time is given to the nearest second, and the rate is the number of participants who completed the task in the allotted time.

Platform	Task					
	Nanotube		Helicene		Knot	
	t	p	t	p	t	p
Mouse	-3.320	0.002	-0.746	0.462	-1.06	0.45
Touchscreen	-2.254	0.038	0.003	0.998	n/a	n/a

Table 5.2: Table showing the results of Welch's test for the hypotheses that participants using the VR platform have faster completion times than the mouse and touchscreen platforms. For each platform, mouse and keyboard, the t-value and corresponding p-value for each task is shown.

to the other. The Welch's t-test computes the statistic t, which indicates how different the two sample means are, via:

$$t = \frac{\bar{x}_a - \bar{x}_b}{\sqrt{\frac{s_a^2}{N_a} + \frac{s_b^2}{N_b}}},$$

where $\bar{x}_a$, s_a^2, N_a, $\bar{x}_b$, s_b^2 and N_b are the sample mean, sample variance and sample size for samples a and b respectively. The t-value for a particular t-distribution can then be used to compute the p-value, which gives the probability that the difference in the observed sample means is due to chance.

For the nanotube task, the VR platform completion times give a p-value of less than 0.05 against the other platforms, as shown in Table 5.2. For the other platforms and tasks, the differences between the means are not statistically significant, indicating that VR did not give a statistically significant advantage in completion time.

5.2.2 Qualitative Feedback

As well as gathering quantitative data to evaluate the VR framework, the studies included qualitative feedback components to gather some insights as to how the user

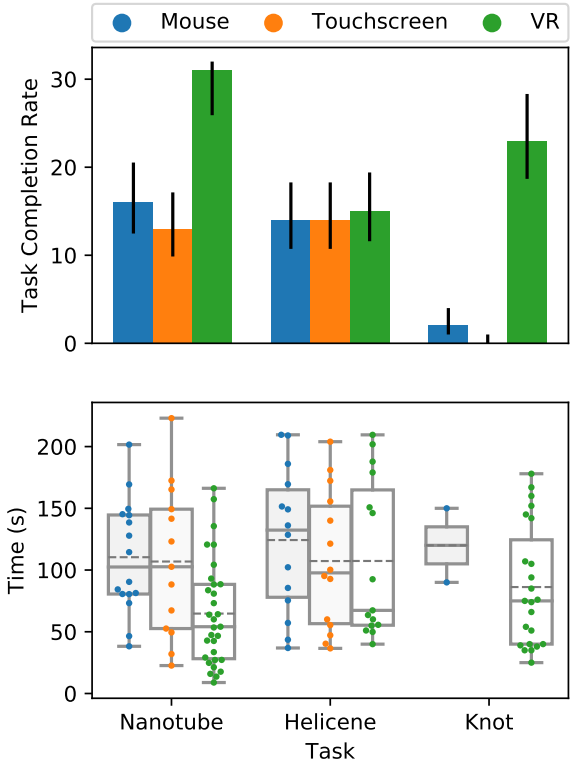


Figure 5.3: The task accomplishment rates and completion times by users in the study. The top panel shows user accomplishment rates for the tasks outlined in Figure 5.1, with error bars produced with Poisson error estimation. The bottom panel shows the corresponding distribution of task accomplishment times, along with box-and-whisker plots. The box ranges from the lower to upper quartile values of the observed completion times, with a solid line at the median. The mean is shown with a dashed line. The whiskers illustrate the full range of the data. The distribution of the data set is overlaid with a swarm plot, where points are adjusted on the horizontal axis so as to not overlap.

experience varied between platforms. In the first user study, a questionnaire was used, while in the second user study, in which participants undertook the knot tying task, participants were interviewed after finishing the tasks. The participant's self-reported familiarity with each platform was recorded on a Likert scale, where 1 represents no experience, and 5 represents a great deal of experience and is shown in Figure 5.4. Generally, prior experience of VR was found to be low, while familiarity with tablet touchscreen interfaces was found to be reasonably high, which is consistent with the market availability of the platforms. Despite such familiarity with the tablet and mouse platforms, participants overwhelmingly preferred the VR interface to the other interfaces (41 out of 44 participants). What was it about the VR interface that made the tasks easier?

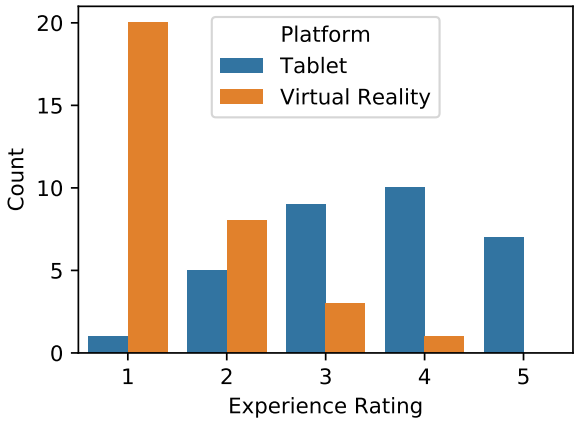


Figure 5.4: Likert scale responses indicating participants self-reported familiarity with the VR and tablet platforms.The scale ranges from 1 to 5, with 1 indicating no familiarity and 5 indicated very high familiarity. High familiarity with the mouse and keyboard platform was assumed.

The Likert scale responses on the features of the VR interface are shown in Figure 5.5. The general trend was that all three features, depth perception, physical movement and the ability to interact with two atoms simultaneously were all useful in some way, but participants seem to agree that depth perception was the most important feature.

In feedback, participants explicitly commented on the difficulties posed by the mouse and tablet interfaces in not being able to rotate the view and grab atoms at the same time and observed that it was difficult to determine where in 3D space the plane of interaction (see Figure 5.2) was moving atoms. This was particularly problematic for the knot-tying task, which requires creating a loop through which to thread an end of the peptide. Because the molecular dynamics was continually running, if a participant had to stop interacting with an atom to rotate the camera it often meant that some progress towards tying the knot was lost. In contrast, in VR it was easier to choose atoms to interact with and guide them more precisely, while simultaneously adjusting perspective. Additionally, different parts of the molecule could be held simultaneously with the two controllers, providing avenues for anchoring a section while pulling an end through to thread the loop.

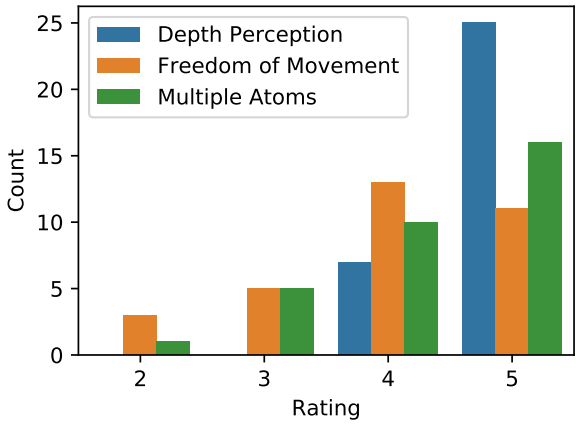


Figure 5.5: The Likert scale responses indicating participants opinions on the importance of various aspects of the VR interface. The scale ranges from 1 to 5, with 1 indicating the feature is of no importance and five indicating that the feature is very important.

5.3 Conclusions

The user study on task accomplishment provided an initial benchmark and evaluation of the effectiveness of virtual reality for the manipulation of molecular conformations. The virtual reality interface was found to be the most effective platform for completing the tasks, with the efficiency gains increasing with the amount of 3D manipulation required in the task. For the most complex task, tying a knot in a small peptide, users were an order of magnitude more likely to be able to complete the task with the virtual reality platform compared to other platforms. Additionally, despite generally low familiarity with VR interfaces, users overwhelmingly preferred interacting with the molecular system using the VR interface. The study provides evidence that the IMD framework using VR enables efficient exploration of molecular configurations, supporting the notion that it could be a useful method for generating pathways and conformational hypotheses.

With the resources available, it was only possible to compare the VR platform to mouse and touchscreen interfaces, and user feedback indicated a variety of features of the VR interface that made it excel. There are other 3D interfaces that one could compare against which could isolate the importance of these features. In the context of interactive molecular dynamics, an obvious example is the haptic devices used in previous IMD implementations[153]. Such devices are prohibitively expensive and so were not available for this task (a key advantage of commodity virtual reality hardware). However, unlike the mouse/touchscreen interfaces, this interface does allow for

3D control but is not co-located, and so would provide an insightful indication as to how much co-located input devices within the virtual reality environment contribute to task accomplishment.

The lower-cost virtual reality platforms such as the Google Cardboard or Gear VR do not provide six degree-of-freedom tracking, and controllers are not tracked in the virtual environment. Repeating the study with one of these devices would isolate the effect of depth perception, enabling one to test how important this feature is. Additionally, these low-cost virtual reality platforms are the most widespread, so determining whether these interfaces are sufficient for manipulation of molecular systems would be beneficial. Similarly, mixed reality platforms such as the Hololens, which rely on gesture controls to interact with virtual objects, would also be an interesting comparison point.

Based on user feedback and extensive personal experience with all these platforms, it is the opinion of the author that these other platforms (in their current forms), would not be as effective the VR platform. It is the combination of all the features of VR - the depth perception provided by the HMD, the co-located controllers and high-precision tracking allowing freedom of movement - that give VR the advantage. With all these features combined, the feeling of immersion transcends into the experience of *presence*[181], in which one feels as if they are embodied within the virtual environment. The sense of proprioception enables the controllers to become an extension of oneself, and thus the simulated model becomes tangible, with the user able to, for example, hold one section of a molecular system in place while looking at another. Combined with accurate depth perception, it becomes much easier to perform precise manipulations.

A criticism of the study is that it does not address the accessibility of the other platforms compared to VR. While VR may provide useful benefits in completing a task, are they enough to outweigh the ubiquity of mouse and keyboard? The commoditization of VR technology means that the barrier to access of this equipment is continuously decreasing as the technology gets cheaper and more widespread. However, like any technology, a combination of market forces and 'killer' applications must be demonstrated in order for it to be adopted. In the author's opinion, the complexity of some of the tasks that molecular scientists have to achieve is sufficient for there to be a case for the adoption of VR in computational chemistry labs.

The tasks designed for the user study, as well as being interesting toy systems analogous to true applications, provided a basis on which to evaluate the framework from end-to-end in both user experience, task accomplishment and technical stability. With

a complex software and hardware stack from GPU-accelerated simulations running on commercial cloud through to virtual reality user interfaces available to the public for free, the technical accomplishment achieved through collaboration across a range of fields is notable in its own right. Follow up user studies investigating the accomplishment of tasks of real use to computational chemistry workflows, such as interactive drug-binding to enzymes, are currently being undertaken.

While the study evaluated the accomplishment of tasks, it did not evaluate the utility of the pathways found by users, and no quantitative data such as collective variables or free energies were extracted. In the following chapters, methods for evaluating trajectories generating VR, and extracting pathways, collective variables and seeds from which to run additional sampling from the interactive molecular are explored in a variety of applications.

CHAPTER 6

GENERATION OF DYNAMICAL PATHWAYS ON ALANINE DIPEPTIDE

In the previous chapter, various reconfigurations of molecular systems were presented as tasks for users to achieve. The reconfigurations, transitions, or reactions between states in a molecular system are often considered in terms of dynamical pathways. These pathways are used as both a conceptual description of a dynamical process, a convenient dimensionality reduction, and as tools in analysis and sampling of these processes. In the study of chemical transformations, when one is interested in determining the transition state of the process, it is common to optimise some initial path with path optimisation methods. There are many such methods, including the nudged elastic band (NEB), the string method, the growing string method, and others[169, 215, 216]. Additionally, such paths may be used as a basis for additional sampling, as in transition path sampling, transition interface sampling, or metadynamics with path collective variables[95, 129].

To use these methods, however, one needs an initial path. For small systems, given some definition of start and end points, interpolation methods are often used to provide these pathways[171]. However, this approach means the path may traverse through unphysical regions of phase space. For example, clashes between atoms that pass through each other may occur. Such pathways are then difficult to optimise, getting trapped in unrealistic configurations, and can require complex optimisation methods to resolve, such as the quasi-continuous interpolation scheme described by Wales *et al*[171]. As

the size of the system increases, the curse of dimensionality rears its head, with optimisation methods becoming increasingly unlikely to converge to optimal pathways. Additionally, these methods presuppose that one has a reasonable start and end state, which may not always be the case.

Another approach is to generate pathways from molecular dynamics simulations. The challenge then becomes overcoming the rare event problem in order to sample the event of interest. High temperature or high pressure simulations, or simulations biased to propagate along some collective variable hypothesis may provide these initial pathways[129, 217]. Essentially, one seeks to find a reasonable initial path by any means possible, and optimise it in post-processing.

The interactive molecular dynamics framework described in the previous chapters could be a useful tool in the arsenal for finding these paths. In this chapter, the iMD-VR framework is applied to the task of finding an initial pathway in the isomerisation of alanine dipeptide. This pathway is evaluated, then optimised with the nudged elastic band method and used as a basis for additional sampling with metadynamics. The purpose of this study is to evaluate the interactive molecular dynamics workflow and how it interacts with existing path methods. The full set of input files, analysis scripts and data for this chapter are available in Ref [218].

6.1 Pathways Generated in Virtual Reality

The isomerisation of alanine dipeptide is a staple benchmark used to test path sampling, rare event and CV analysis methods[129, 134, 219–225]. In vacuum, the states C_{7eq} and C_{7ax} are prominent, as shown in Figure 6.1a, and transitions between these states can be characterised with paths using the dihedral angles ϕ and ψ as collective variables[221, 224], as shown in the free energy surface in Figure 6.1b. This free energy surface was computed from a single one nanosecond metadynamics trajectory at 300K using OpenMM using the AMBER99SB force field under Langevin dynamics with a time step of two femtoseconds. The trajectory was biased with metadynamics along both the ϕ and ψ angles. Standard metadynamics was used, with Gaussians of a height of 1.2 kJ/mol and standard deviation of 0.35 radians for both ϕ and ψ deposited every 500 molecular dynamics steps. The resulting free energy surface, with a barrier of 8 kcal/mol between the C_{7eq} and C_{7ax} states, is in good agreement with previous calculations[129, 225], sufficient for the following discussion.

The system was simulated in iMD-VR analogously to the metadynamics calculation,

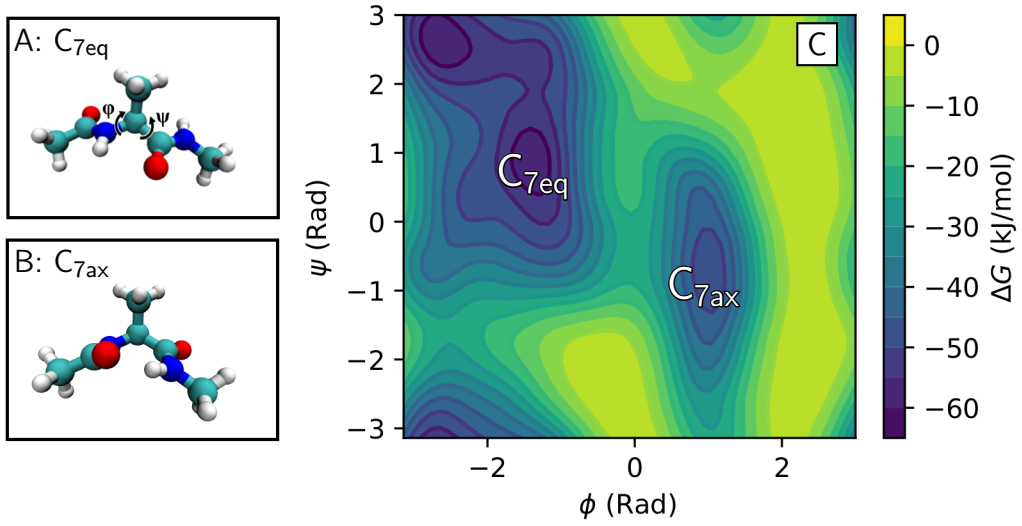
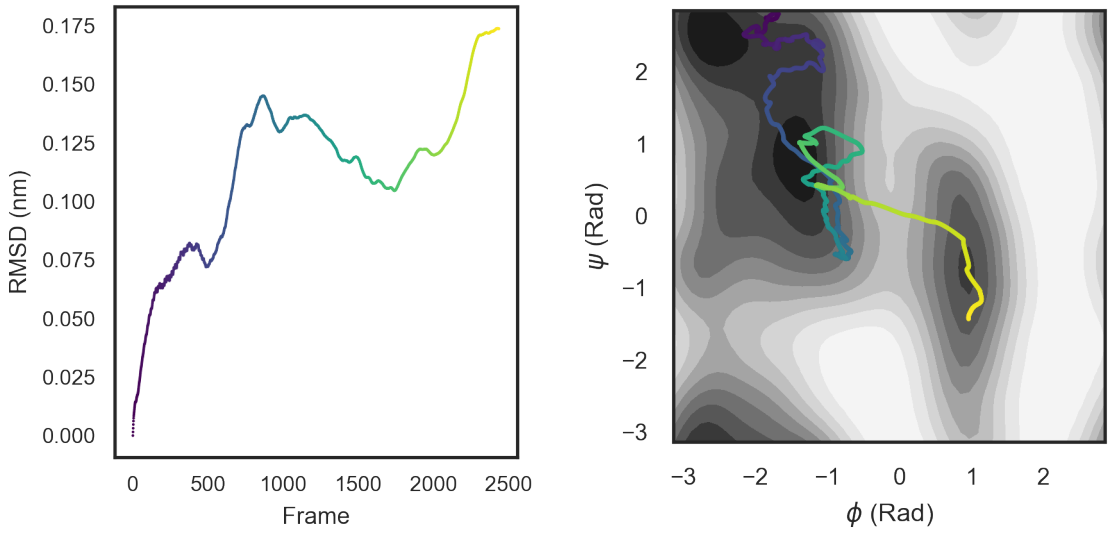


Figure 6.1: Isomerisation of alanine dipeptide: (A) Shows the C_{7eq} state, (B) shows the C_{7ax} state, and (C) shows a free energy surface as a function of the dihedral angles ϕ and ψ calculated using metadynamics. Isolines are drawn every 5 kJ/mol.

using the OpenMM plug-in and AMBER99SB force field, with a Velocity Verlet integrator and Berendsen Thermostat set to 300K. A time step of 0.5 fs was used to provide greater control as it is a relatively fast transition. The Gaussian interactive potential was used, with the scaling parameter c set to 1000 kJ/(mol*a.m.u.). A restraining potential, as described in Chapter 4.2, was applied to the central carbon atom so the user could focus on the rotation. The system was then manipulated through the application of the bias potentials, attempting to transition smoothly from state C_{7eq} to C_{7ax}. The first attempt with these settings resulted in a successful transition carried out over a 300 fs period.

With a pathway generated from an interactive session, the next step is to process it into a simple path. A challenge here for a general molecular system is to find appropriate representations that allow for the identification of the key configurations in the pathway. One of the goals for the IMD pipeline is for it to be as generic and ideally black-box as possible, as discussed in Chapter 4.2 (see Figure 4.1). A simple general metric that can be applied to many systems is the root mean square deviation (RMSD), which is shown for the IMD trajectory in Figure 6.2. The RMSD was computed with MDTraj[49], using the initial condition as the reference frame. The RMSD (or MSD) is often used as a way of measuring the distance between configurations in pathways, and in what follows, is even used to bias the system along the pathway. However, for



(a) (b)

Figure 6.2: The raw molecular dynamics trajectory generated with interactive molecular dynamics. A) RMSD of each frame with respect to the first frame. B) The trajectory projected onto the dihedral angles ϕ and ψ.

a system with many degrees of freedom it can be a misleading metric in constructing a pathway. Indeed, even in the small system of alanine dipeptide, other pathways such as a complete rotation about the ψ angle would confuse this metric. For the benchmark system of alanine dipeptide, however, we can leverage the knowledge of two key collective variables ϕ and ψ to characterise the pathway. The dynamical pathway generated during this interactive session is shown in Figure 6.2b, represented in terms of the dihedral angles ϕ and ψ, with the free energy surface computed from the metadynamics displayed for reference. There are a few observations to be made about this path. Firstly it follows the minimum energy pathway reasonably well, which makes it a good candidate for use as a path to accelerate dynamics. However, as the path is generated from molecular dynamics, the path is full of vibrations, turns and loops as random (and user-guided) interactions push the dynamics around. A smoother, simplified path is desired to serve as a base for optimisation. In this case, it was a trivial exercise to select some representative frames along the trajectory by hand. 21 points along the trajectory were used to form the initial pathway, and are shown in Figure 6.3.

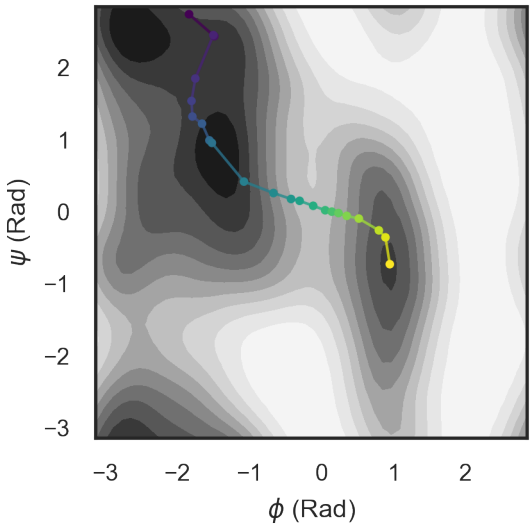


Figure 6.3: The alanine dipeptide isomerisation trajectory generated interactively using iMD-VR.

6.2 Nudged Elastic Band Optimisations

With a candidate path in hand, one typically seeks to locally optimise the pathway to provide more accurate details of the energetics along the path, or for use in follow-up sampling. The candidate path produced for the alanine dipeptide system was optimised using the nudged elastic band algorithm[170], with the Atomic Simulation Environment (ASE) python package[226]. Given an initial starting path consisting of a set of configurations, or 'beads', the method optimises a path on the potential energy surface by relaxing the beads, with a local optimisation algorithm, based on the forces from the potential energy surface, with spring forces between the beads used to preserve the overall integrity of the path. The details of the method can found in Refs [169, 170]. An OpenMM plug-in to ASE was developed to enable the same forcefield as that used in the previous sections to provide energy calculations and gradients[1]. The improved tangent NEB method provided by ASE was used, with a force constant value, k, of 0.5 eV/Å, with rotations and translations in the molecule removed. A BFGS optimiser, which belongs to the quasi-Newton family of optimisation methods[226], in which approximate calculations of the Hessian matrix (the matrix of second order derivatives) are used to perform local optimisation, was used. The improved tangent NEB method, as its name suggests, provides an improvement to the definition of the tangent between each bead,

[1]Code available at *bitbucket.org/mikeoconnor0308/openmm-ase-calculator*.

which is used to define the orthogonal direction along which forces from the underlying potential energy surface are applied. A convergence criterion of a maximum force of 0.04 eV/Å between any pairs of beads was used. Before performing the NEB, the start and end points were minimised using a BFGS local optimisation, as they are fixed during the NEB calculation. The relaxed path after NEB is shown in Figure 6.4a, and took 277 steps to converge. The optimised pathway now smoothly transitions between the energy wells and follows the minimum energy pathway. While a full transition state search was not performed, the barrier height of approximately 8 kcal/mol is consistent with previous results[225]. There is, however, some slipping of an additional bead at the start and end point into the wells.

To provide a comparison of the interactive pathway to the off-the-shelf methods available in ASE, the same NEB calculation was performed using both a simple interpolation between the start and end points as an initial guess. The interpolation method failed to converge for any values of a range of values k, the force constant, between 0.1 and 50, with nonsensical paths emerging. It was necessary to change the optimizer to the FIRE variant, which is an optimizer drawing inspiration from molecular dynamics[216], after which the interpolation method did converge after 298 steps at a force constant value of 0.5 eV/Å. However, the resulting path does not follow the minimum energy pathway, as shown in Figures 6.4b and c.

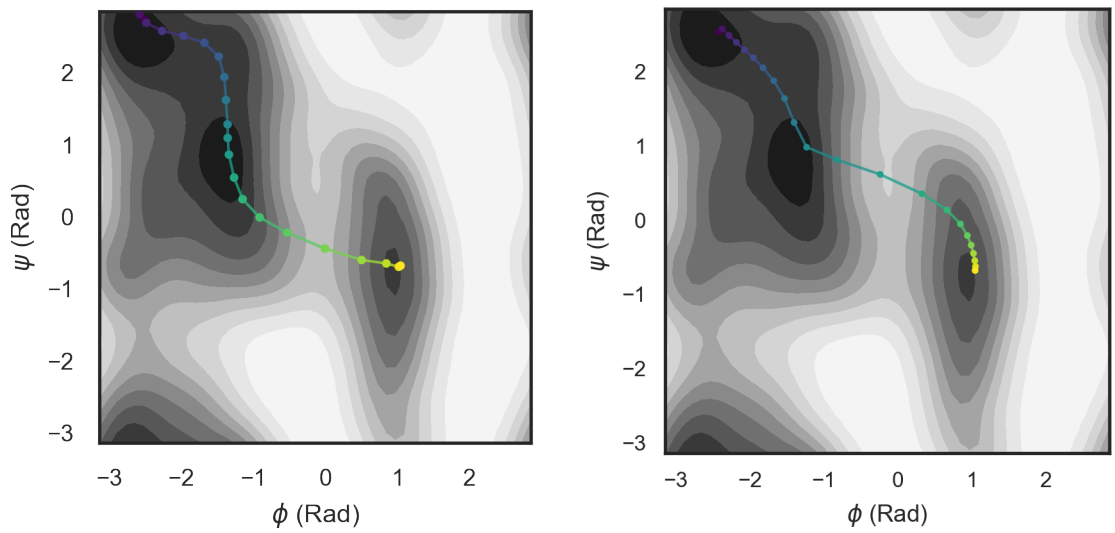
6.3 The Path Collective Variable

The optimised path produced by the nudged elastic band method on the potential energy surface characterises the reaction pathway but provides only enthalpic information. If we wish to sample entropic effects, additional sampling is required. It would be useful to have a measure of where on the path a configuration is, as well as how much it has diverged from the path. We may also want to accelerate dynamics along the path to converge a free energy surface. A method for achieving this proposed by Branduardi *et al* is the path collective variable[129]. The path variable takes the following form:

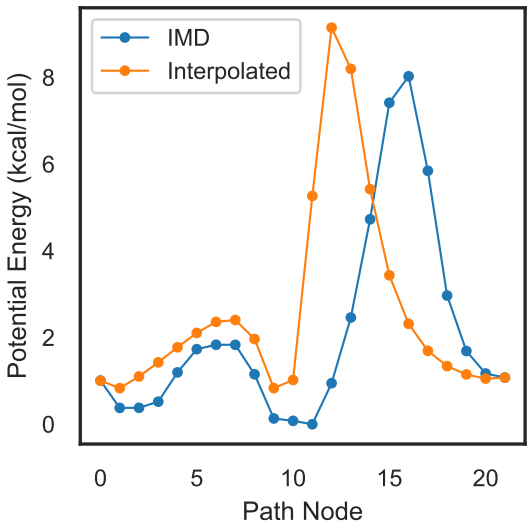
$$S(\mathbf{R}) = \frac{\sum_{i=1}^{N} i \exp(-\lambda |\mathbf{R} - \mathbf{R}_i|)}{\sum_{i=1}^{N} \exp(-\lambda |\mathbf{R} - \mathbf{R}_i|)}$$

where $\mathbf{R}$ represents a given configuration, $\mathbf{R}_i$ is a sequence of N configurations that represents the path, and $|\mathbf{R} - \mathbf{R}_i|$ is some metric of difference between configurations.

When a configuration is close to the configuration $\mathbf{R}_i$ on the path, the value $|\mathbf{R} - \mathbf{R}_i|$ will be close to zero, and the other terms will vanish if λ is sufficiently large, so $S(\mathbf{R})$



(a) (b)



(c)

Figure 6.4: The paths optimised with the nudged elastic band method. A) The IMD path after optimisation plotted as a function of dihedral angles of alanine dipeptide, B) the path produced by linear interpolation, using the FIRE optimiser, after optimisation. C) Comparison of the potential energy along the pathway found with IMD compared to the pathway found with linear interpolation.

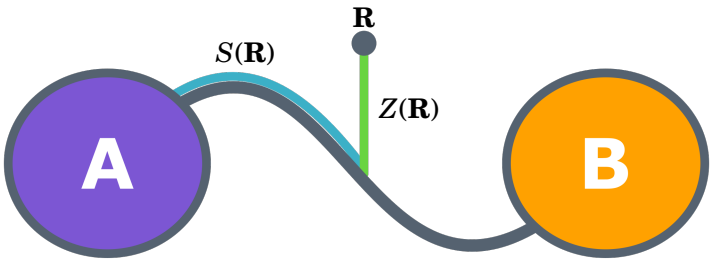


Figure 6.5: Schematic of the path collective variable between two states A and B.

will tend to the value i. The parameter λ provides smoothing between each configuration. For the interpretation of $S(\mathbf{R})$ to smooth, continuous and meaningful between the values of 0 and N, the following requirements must be met. The distances between adjacent configurations on the path, $|\mathbf{R_{i+1}} - \mathbf{R}_i|$, must be as close to equal as possible, so each integer value of the path represents a similar distance in configuration space. Additionally, λ must be set so that values of $S(\mathbf{R})$ smoothly transition between the integer values along the path. If it is set too low, $S(\mathbf{R})$ will not distinguish between configurations. If it is set too high, $S(\mathbf{R})$ will not smoothly transition between configurations, and instead jump to near-integer values. The suggested rule of thumb is to set λ to be proportional to the inverse of the mean square displacement between successive frames[227].

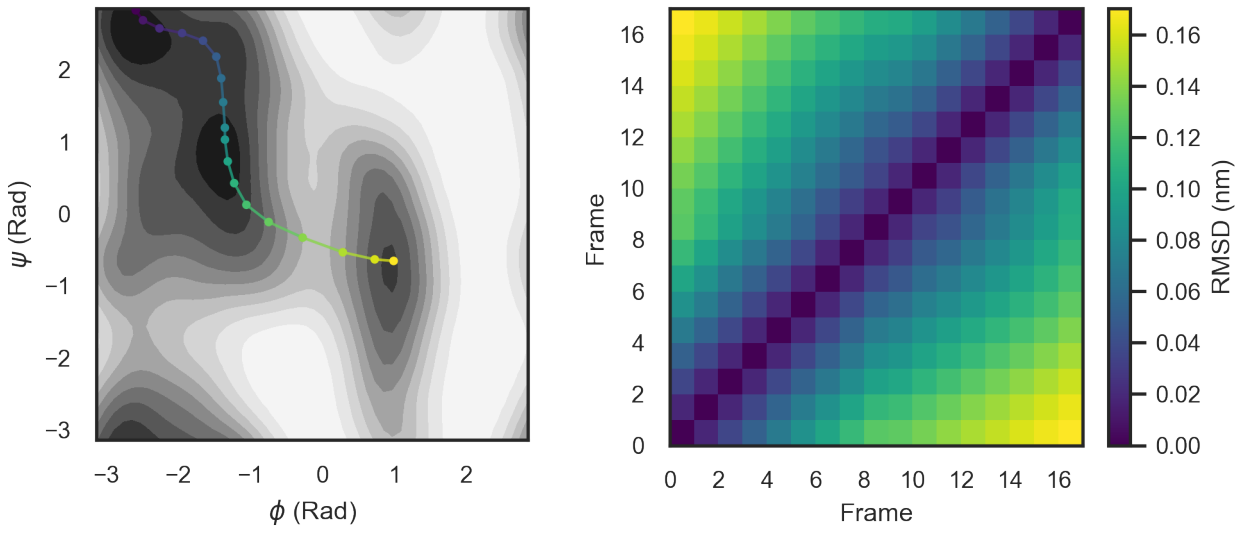
The value $S(\mathbf{R})$ indicates how far along the path we are, but it tells us nothing of how far from the path we are (see Figure 6.5). An approximate measure of the distance from the path is given by

$$Z(\mathbf{R}) = -\frac{1}{\lambda}\log\left(\sum_{i=1}^{N}\exp\left(-\lambda\,|\mathbf{R}-\mathbf{R}_i|\right)\right).$$

The values $Z(\mathbf{R}) = z$ form a tube around the path with radius z. This metric enables one to discover alternative pathways to that defined by the original path. Typically the RMSD or MSD distance is used as a distance metric $|\mathbf{R}-\mathbf{R}_i|$, but other measures such as the difference of appropriate collective variables could be used.

With such a definition, the path collective variable is smooth and differentiable, and so can be used as a collective variable for a biased simulation with an enhanced sampling method such as umbrella sampling, metadynamics or BXD.

The optimised path produced by the nudged elastic band method was configured for use with the path collective variable using the tools provided by the PLUMED package, which, given an initial path, adjust the positions of the nodes along the path such that they are equidistant from one another. The two beads from the optimised path that had



(a) (b)

Figure 6.6: The processed path for use with the path collective variable. A) Projection of the path onto the dihedral angles ϕ and ψ of alanine dipeptide. B) The RMSD matrix between all pairs of configurations on the path, coloured from blue to yellow by increasing RMSD.

fallen into the wells were removed for convenience. The resulting path is shown in Figure 6.6a. A useful visualisation to verify that the path is sufficiently smooth for use with the path collective variable is the matrix of pairwise RMSD calculations between configurations on the path. The flatter the shape of this surface, especially on the diagonal elements which represent the pairwise distance between configurations on the path, the better, as it indicates the nodes along the path are equidistant from one another[129] and monotonically increasing in RMSD. The path for alanine dipeptide exhibits good properties for use with the path CV, as it is smoothly increasing. With an average distance between configurations of 0.018 nm and a standard deviation of 9.145e-06 nm, the path configurations are very close to equidistant.

6.4 Metadynamics on the Path

The last step in the pipeline proposed in Chapter 4.2 from an iMD-VR session to observables such as free energies is additional sampling based on configurations found in the session. This final step was explored by running metadynamics using the path collective variable with the path optimised in the previous section. This section replicates the

method of one of the metadynamics tutorials provided by the PLUMED developers[228], but using a path independently generated with iMD-VR rather than the one supplied, and operating on the Amber99SB forcefield in OpenMM. The well-tempered variant of metadynamics with adaptive Gaussians was used, which automatically adjusts the height and shape of Gaussians deposited over time, accelerating convergence and reducing the effect of parameter choice on the results[116, 229]. In the well-tempered adaptive Gaussian metadynamics variant, one needs to set the length of time over which to sample in order to define the Gaussian sizes, τ_D. This was set to every 100 steps. Additionally, one needs to specify the initial Gaussian height, which was set to 3.2 kJ/mol. The bias factor, which adjusts the speed at which the Gaussian heights decrease, was set to 10, and the pace at which Gaussians are deposited which was set to every 125 steps. These parameters were based on the typical inputs used in existing examples of metadynamics calculations on alanine dipeptide[115]. A metadynamics trajectory of 5 ns was run, in the same manner as the original metadynamics calculation described above in which the dihedral angles were used as the collective variables.

Resulting free energy surfaces were computed with the histogram re-weighting procedure described in Ref [229], using the PLUMED command line tools.

6.4.1 Results

Figure 6.7 shows the resulting 2D free energy surface in terms of the components of the path collective variable $S(\mathbf{R})$ and $Z(\mathbf{R})$. A number of heuristics can be applied to assess the convergence of the free energy surface, and are shown in Figure 6.8. Firstly, one must ensure that the expected range of collective variable space has been observed multiple times. This is shown in panels A and B of Figure 6.8, and it is clear that the full path has been repeatedly sampled, as well as regions distant from the path. The trajectories are not completely diffusive across the collective variable space, however, indicating that more sampling may be required for full convergence.

Another heuristic, the height of the Gaussians deposited over the course of the trajectory, indicates convergence, as the Gaussians decrease dramatically in height to a median of 0.05 kJ/mol in the last 100 depositions of the trajectory. The exceptions to these small deposits, indicated by large spikes in the plot, correlate with large values of $Z(\mathbf{R})$, which correspond to regions of collective variable space far from the path, and so are not of great concern. Finally, panel D of Figure 6.8 shows the convergence of a 1D projection of the free energy surface along $S(\mathbf{R})$ over time. This was produced by computing cumulative free energy surfaces after every 500 Gaussian depositions, indicated

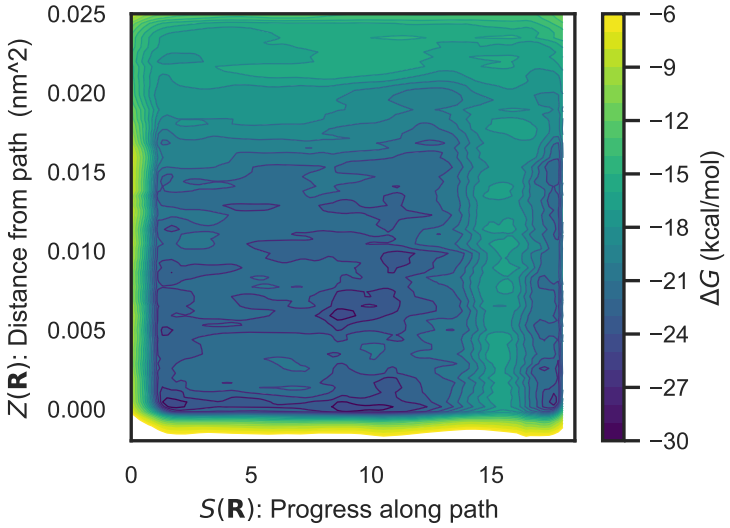


Figure 6.7: Free energy surface computed using well tempered metadynamics on the path collective variable. Isolines represent a free energy difference of 1 kcal/mol.

by decreasing transparency and darker shades of blue. The 1D projection is computed by integrating out the second collective variable $Z(\mathbf{R})$, and is presented only to assess convergence. The final free energy surface is shown as the solid grey line. The free energy surface appears to converge rapidly, not changing significantly after the first few strides. Taking these heuristics in combination, it can be concluded that the free energy surface depicted in Figure 6.7 is sufficiently well converged for this discussion.

However, the free energy surface in Figure 6.7 produced through the use of the path collective variable can be confusing and difficult to interpret. It is helpful to visualise the trajectory projected onto other collective variables, in this case, the ever-reliable dihedral angles ϕ and ψ, to interpret the surface. These projections are shown in Figure 6.9. The first two panels of this Figure, A and B, show the value of $S(\mathbf{R})$ and $Z(\mathbf{R})$ in terms of the angles ϕ and ψ. From this, we can see that the path variable behaves as one would expect, with $S(\mathbf{R})$ smoothly transitioning along the path, with values of $Z(\mathbf{R})$ near zero corresponding to the positions very near the pathway. The free energy surface of Figure 6.7 is hence straightforward to interpret in this region, for values of $Z(\mathbf{R})$ near zero, the shape is clearly reminiscent of the surface one expects, with three minima, and a large barrier between C_{eq} and C_{ax}. The barrier height is approximately 8 kcal/mol, in good agreement with the barrier calculated by metadynamics along the ϕ and ψ angles presented in Figure 6.1 and previous calculations. Up to a $Z(\mathbf{R})$ value of around 0.015 nm^2, the corresponding ϕ and ψ angles (shown in panels C and D of Figure 6.9) are

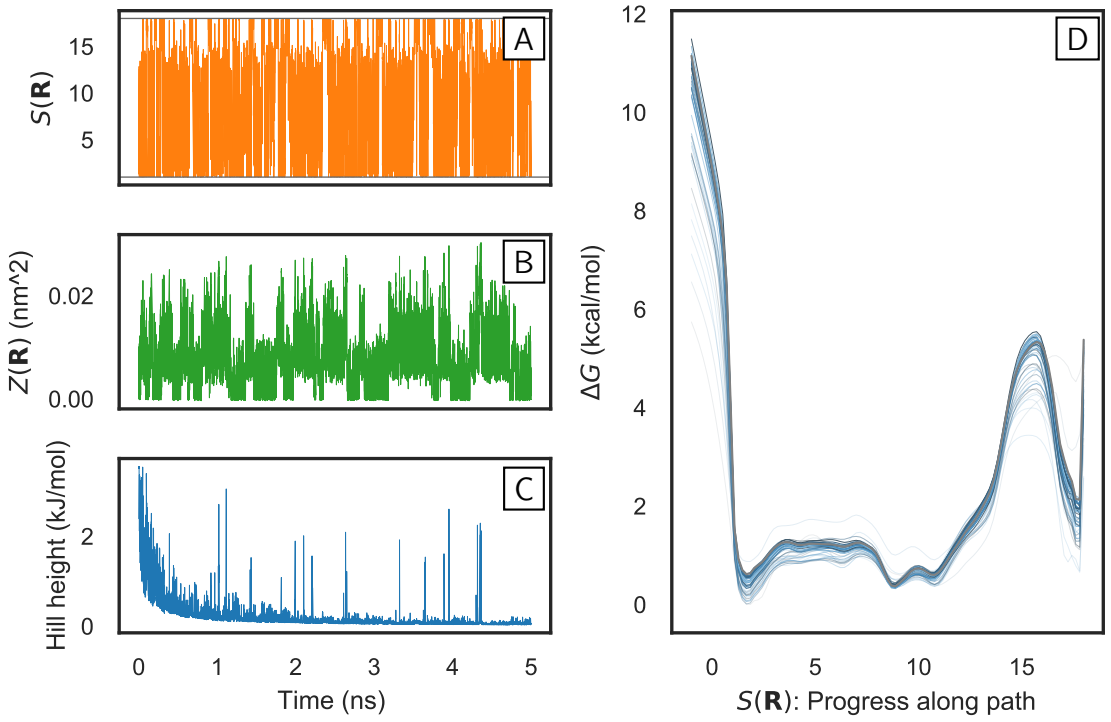


Figure 6.8: Heuristics to assess the convergence of the metadynamics trajectory. A) The value of $S(\mathbf{R})$, the progress along the path, over the course of the trajectory. B) The value of $Z(\mathbf{R})$, the distance from the path, over the course of the trajectory. C) The height of the deposited Gaussians over the course of the trajectory. D) The convergence of a 1D projection of the free energy surface in terms of $S(\mathbf{R})$ as a function of increasing trajectory time, at a stride of 0.1 ns. Lighter and more transparent shades of blue indicate the earlier free energy surface estimates, while the solid grey line indicates the final free energy surface.

consistent with the structure of the path. Beyond this distance, the free energy surface becomes difficult to interpret as the path variable begins to distort across additional pathways and the periodic nature of the system. In particular, the rotation about ψ between 0 and -2 radians, with ϕ approximately -3 radians, corresponds to many values of $S(\mathbf{R})$.

An additional observation to make is the distortion of the free energy surface at the endpoints of the path. At these points, the collective variable becomes discontinuous as the values of $S(\mathbf{R})$ of 0 and 17 are hard limits, causing artefacts in the histogram procedure. One way to alleviate this is to include additional points along the path beyond the start and end points of interest.

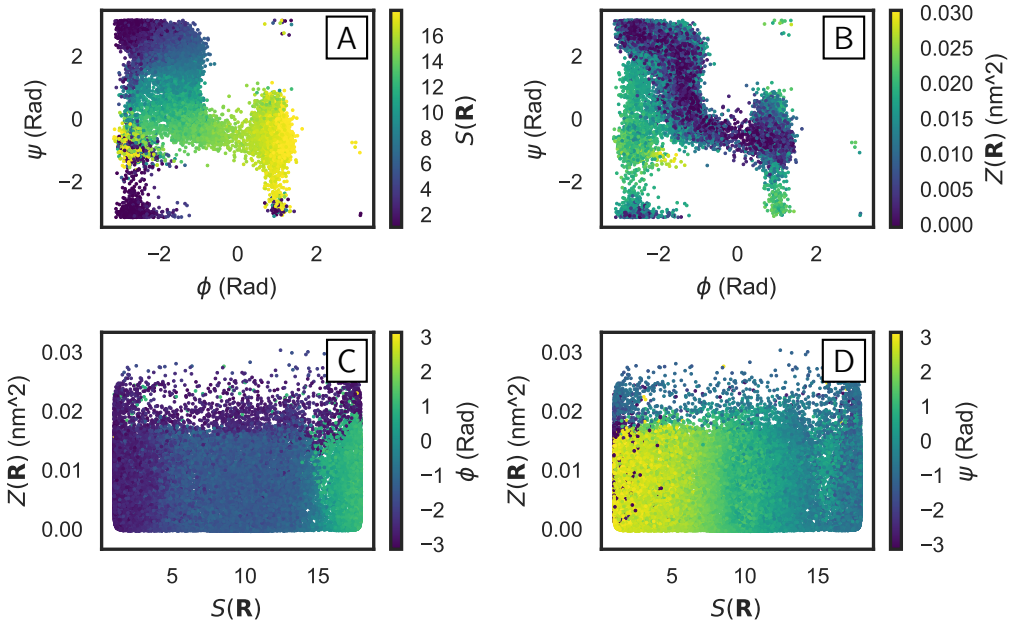


Figure 6.9: Projections of the metadynamics trajectory of the alanine dipeptide system onto the collective variables $S(\mathbf{R})$, $Z(\mathbf{R})$, ϕ and ψ. Projection of the trajectory onto A) the angles ϕ and ψ, coloured by distance along the path $S(\mathbf{R})$, B) the angles ϕ and ψ coloured by distance from the path $Z(\mathbf{R})$, C) the path variables $S(\mathbf{R})$ and $Z(\mathbf{R})$, coloured by the angle ϕ, D) the path variables $S(\mathbf{R})$ and $Z(\mathbf{R})$, coloured by the angle ψ.

6.5 Discussion and Conclusions

This chapter provides a demonstration of a workflow using interactive molecular dynamics sessions to sample pathways, which can then be used to accelerate sampling and converge observables. The use of the benchmark system alanine dipeptide allowed for critical evaluation of the methods as the pathways produced could be validated.

The interactive molecular dynamics framework made creating an initial sample of the isomerisation of alanine dipeptide trivial, and it was shown that despite the large biasing potentials applied during an interactive session, the resulting trajectory followed a pathway close to the minimum energy path. In this work, the raw molecular dynamics trajectory was analysed, and a pathway of a small number of configurations was extracted by hand. In order to make the workflow more black-box, towards the ideal outlined in Chapter 4.2, this process would ideally be automated. Wang *et al* have described a process, coined *nebterpolation*, for automatically extracting reaction pathways from *ab initio* trajectories[217] which includes a filter applied to the internal coordinates of the system to produce a smooth pathway that can then be used as a path for

optimisation. While the method specifically targets reaction pathways, it may be applicable or extendible to conformational changes in larger biomolecular systems treated with the molecular mechanics forcefields used here.

The initial pathway produced from interactive molecular dynamics was optimised on the potential energy surface with the nudged elastic band method. By providing a pathway that was already reasonably near the minimum energy pathway as a starting point, NEB quickly converged to a reasonable path using default parameters. The fact that so many different parameters had to be trialled in order to converge NEB with ASE using interpolation to a passable pathway, for such a simple system, provides an indication of the difficulties users face in tuning hyper-parameters[2] to find reasonable pathways in complex systems. However, this appears to be an implementation issue, as there are many examples of successful NEB calculations of alanine dipeptide with other software packages[231, 232]. As a package primarily used in materials and electronic structure calculations, it is possible that the implementation in ASE is not tuned for molecular mechanics force fields, which can exhibit high energies due to the use of spring potentials. Indeed, example calculations on alanine dipeptide using the molecular mechanics program AMBER make use of simulated annealing and a slow heating process to ensure stability[233]. Applying similarly careful strategies to the calculation performed with ASE may yield better results with interpolation. Nevertheless, these considerations further highlight the utility of pathways produced from molecular dynamics, and in particular interactive molecular dynamics, which has been demonstrated here to efficiently produce a reasonable initial pathway which can be optimized effectively.

The pathway was then optimised for use with the path collective variable and accelerated with metadynamics using the resulting path. The resulting free energy surface was reasonably converged to the qualitative level required for this study and in agreement with a control calculation performed on the dihedral angles. Despite being optimised on the potential energy surface, the path proved to be near optimal in the free energy surface too. Methods for optimising paths used for the path collective variable in free energy space throughout a metadynamics trajectory have been previously demonstrated[129] and are now available in PLUMED 2.4, but an implementation was not available at the time this experiment was carried out.

The path collective variable is a useful way of accelerating dynamics without having to identify more traditional collective variables, but it is also quite esoteric, difficult to interpret without invoking additional collective variables and, like NEB, has a number

[2] An optimization methodology humorously referred to as *graduate student descent*[230].

of parameters to be tuned. These include the λ variable, the distance between each node in the path, the number of nodes in the path, and of course the positioning of the nodes themselves.

As noted, a more recent implementation of PLUMED now includes an adaptive path optimisation procedure which may alleviate some of these issues. Additionally, a new formulation of a path collective variable by Leines and Ensing[146], also available in PLUMED 2.4, claims to provide faster convergence as well as better scaling.

Despite some difficulties in using these methods, this chapter provides evidence that interactive molecular dynamics can provide initial pathways and configurations that can be used to complement existing optimisation and sampling methods, enabling more efficient sampling. Furthermore, the interactive molecular dynamics is simply the starting point in this pipeline, and so can be adapted to any of a number of alternative strategies as they continue to emerge and develop. This idea is explored further in the following chapter, in which Markov state models are used to sample knot formation in the protein MJ0366 and loop motion in the enzyme cyclophilin A.

CHAPTER 7

INTERACTIVE SAMPLING OF PROTEIN DYNAMICS

The previous chapter demonstrated a workflow for pathway discovery and rare event acceleration using interactive molecular dynamics, using the interactive sessions to generate initial pathways of isomerisation in alanine dipeptide that were then used by the path metadynamics method to produce free energy profiles. This is just one of many potential ways in which interactive molecular dynamics can be followed up by other methods to converge statistics. In this chapter, an exploration using the interactive molecular dynamics framework for rare events in considerably larger systems, the knotted protein MJ0366 and the enzyme cyclophilin A, is undertaken, this time using adaptive sampling with Markov state models to perform additional sampling.

7.1 Sampling of Protein Knotting Pathways

The discovery of knotted proteins, starting with some initial findings of simple trefoil knots[234] before the discovery of hundreds more through searches of the Protein Data Bank[212, 235, 236] have become a poorly understood curiosity in the field of protein structure and folding. There have been many experimental studies of these knots, providing insight into possible folding pathways and intermediate structures[237–239]. To understand the folding process, computational studies could provide huge insight. However, due to the timescale limitations of all-atom molecular simulations, and the rare event problem, computational studies have been restricted to simplified coarse-grain models of relatively small proteins[240, 241], where running unbiased molecular dy-

namics fast enough to observe folding events is possible.

This class of proteins make for an interesting use case of the interactive molecular dynamics virtual reality framework described, as the collective motions involved in tying and untying of knots are complex and difficult to accelerate with other methods.

One of the smallest knotted proteins, MJ0366, a hypothetical protein sequenced from *Methanocaldococcus jannaschii*, presents itself as an ideal candidate for exploration. The protein exhibits the simplest knot, a trefoil (3_1) knot[194]. It has been studied computationally using an all-atom coarse-grained model[240], where Noel and coworkers found that knot formation is a late-transition process, and occurs after the main beta-sheet of the protein has formed. They observed there were two pathways to knot formation: 'plugging' and 'slipknotting', as shown in Figure 7.1. In the plugging pathway, the terminal threads through an exposed loop, whereas in the slipknotting pathway the terminal hairpins upon itself and then threads through the loop, creating a slipknot which leads to the final knotted form. Additionally, both slipknotting and plugging of the C-terminal occurred through a loop hereby referred to as the C-loop, occurring 55% and 45% of the time respectively. Intriguingly, when the additional five residues at the C-terminal found in the sequence of MJ0366's crystal structure were included, the slipknotting pathway becomes extremely dominant, occurring 99% of the time, because the plugging pathway has a higher entropic barrier as it needs to line up the C-terminal along the loop.

In addition to the knotting pathways, various kinetically trapped states were found, where the protein forms structure close (in terms of relative atomic positions) to the native state but with malformed knots. In order to progress towards the native state, backtracking must take place to a more unfolded state. The energy landscape is thus rugged and complex, a challenge for molecular simulation.

To date, this system has not been studied using all-atom molecular mechanics forcefields. Using the virtual reality interactive molecular dynamics framework, knotting pathways in this system were explored. The iMD-VR framework was used to perform initial sampling, by untying the protein from its native state, then performing a slipknot or plugging path to re-tie the system. To the author's knowledge, this is the most complex task ever attempted with interactive molecular dynamics. This initial sampling provided the seeds with which to run adaptive sampling using Markov state models.

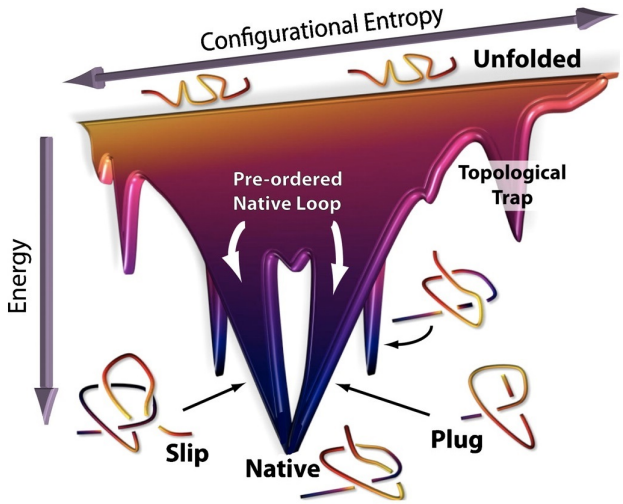


Figure 7.1: Schematic representation of the hypothetical funnel-based energy landscape for the knotted protein MJ0366. The protein folds into an intermediate structure in which the C-loop has formed, after which either the slipknotting or plugging pathway can take place, represented with a bifurcation in the energy landscape. Topological traps, which are close to the native state but could only transition to the native state by significant backtracking, are also shown. Source: [240], used with permission.

7.1.1 Initial Pathway Generation using Interactive Molecular Dynamics

The system was set up to run molecular dynamics in OpenMM, using the Amber10 force field, configured with the PLUMED plug-in described in Chapter 4.2 developed to enable interactive molecular dynamics. The IMD was configured to transmit frames to the client every ten molecular dynamics steps. Starting from the native structure, the system was equilibrated for one nanosecond in 2 nm of an explicit solvent of TIP3P water at a temperature of 300K, using a Langevin integrator with a time step of 2 femtoseconds. A cut-off distance of 2 nm was used for non-bonded interactions, and any bonds involving a hydrogen atom were fixed in length. The simulation was connected to from the iMD-VR client after this time. The positions of all the atoms in the protein were transmitted to the client, but the solvent was not transmitted to reduce bandwidth and rendering costs.

To manipulate the system, any mixture of the selection and interaction tools described were used, as appropriate for the task at hand. To untie the knot from the native state, a selection of the alpha helix that forms the thread through the C-loop was made (residues Ser74-Asp87, the purple section in Figure 7.2). This was then interacted with

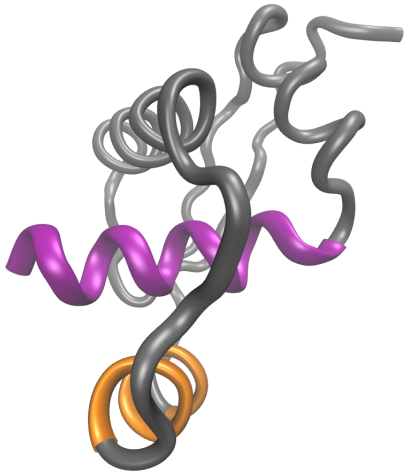


Figure 7.2: The knotted protein MJ0366 in its native structure from PDB entry 2EFV[194], with sections selected for interaction during the interactive molecular dynamics trajectory highlighted. The C-terminal alpha helix, residues Ser74-Asp87, is shown in purple, while the residues Thr43-Glu49 selected from the C-loop are shown in orange.

as a group, applying a force to all atoms equally, to untie the knot. A second selection along some residues of the alpha helix forming the C-loop from residues Thr43-Glu49 was also made (the orange section in Figure 7.2), and this was pulled away from the threading helix, allowing more space for untying. This deformation of this helix was observed in the coarse-grained studies of Noel *et al*[240]. The ability to easily and rapidly select atoms and manipulate them as a group in order to preserve secondary structure as much as possible was crucial for this task. Once the knot has been untied into a partially folded form, both the plugging and the slipknotting pathways were explored. The plugging pathway was performed by selecting the atoms of the C-terminal and pulling them back through the loop, while the slipknot pathway was explored by forming a slipknot by making another selection of atoms at residues Leu78-Asn80 and pulling them back through as a group, as depicted in Figure 7.3.

While performing these tasks, one quickly gains qualitative insights into the system. Untying the knot was an easy task in IMD, and retying with a plug knot was also straightforward. However, one would typically end up with the C-terminus chain under tension as it was pulled through. Tying with a slipknot, as depicted in the fifth panel of Figure 7.3, was challenging, as it is difficult to the manoeuvre the slipknot through the loop, past all the sidechains, without causing significant disruption to C-loop. This is in part due to visual representations used, with the C-terminus chain between residues Ser74 to Asp87 represented with ball-and-stick representation while a ribbon repre-

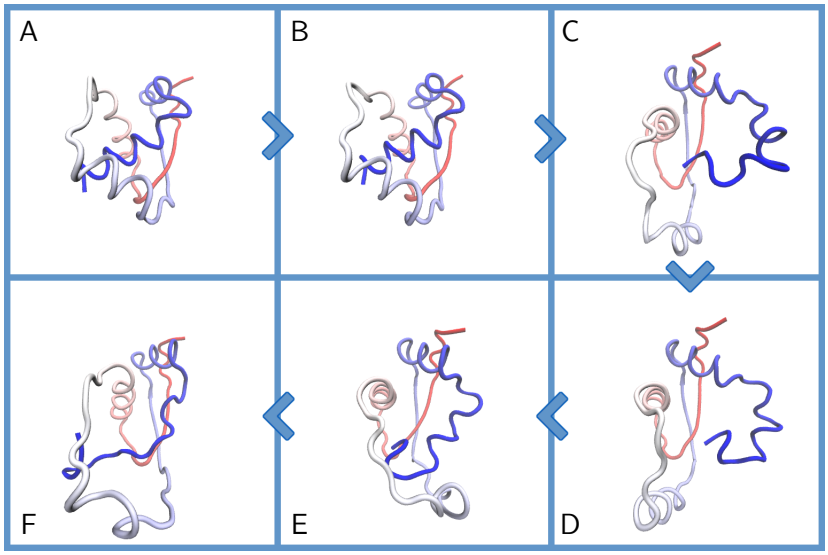


Figure 7.3: Snapshots of an IMD session in which the native state trefoil knot of the protein MJ0366 (panel A) is unthreaded to a partially folded form (panels B and C), and tied with a slipknot to form a loosely knotted conformation (panels D-F)[1].

sentation was used for the rest of the protein, meaning that steric clashes were not always visible. While the user could stop and edit the visualisation parameters to show these regions, it can be cumbersome to do this continually, and the resulting visualisation can become cluttered and overwhelming. An exploration of automated visualisation and sonification to represent such features will be undertaken in future work.

The fact that it seemed to be easier to tie a knot using the plugging motion than the slipknot, which does not match the findings of Noel *et al*, raises a valid concern regarding the use of interactive molecular dynamics for large conformational changes. The user's bias in both the pathways they choose to explore, and the ease with which they can be explored in a high dimensional system within the interactive molecular dynamics framework can have an impact on the qualitative insights gained, and so the need for methods that can test hypotheses generated through IMD is reiterated.

7.1.2 Adaptive Sampling with High Throughput Molecular Dynamics using Markov State Models

Given the application of chemical intuition that has been applied in generating configurations, it is desirable for that to be the final point of human labour in this endeavour, and to pass additional sampling over to automated methods. In the previous chapter, the path sampling strategy using metadynamics and the path collective variable was

explored, but it was not particularly automated, requiring some significant preprocessing and tuning of the path (even for a simple system).

An obvious automated strategy is to simply run unbiased molecular dynamics, using the configurations generated in IMD as initial conditions for sampling. Indeed, this is the approach that was taken in some of the previous applications of IMD[155]. Over the last two decades, the Markov state model (MSM) methodology has been developed for the analysis of molecular dynamics simulations[54, 55]. Combining interactive molecular dynamics with this approach is appealing. As Pande states: “many different types of simulations could be useful in creating the initial data set. One scheme is to ‘seed’ MD simulations, i.e. start them in potentially relevant states apriori"[54]. IMD can readily provide these states, and then one can run dynamics until a converged Markov model based on dynamics from these states is produced.

In practice, however, the unbiased dynamics strategy with Markov models would still suffer from the previously discussed issues of unbiased molecular dynamics - that rare, and typically interesting, events are infrequently sampled. This has of course not gone unnoticed, and hence adaptive sampling strategies based on Markov models have been developed[8, 63, 242, 243] that use the statistical analysis provided by MSMs to improve sampling iteratively. The methodology has seen some high profile success in the atomistic simulation of protein-protein association[104]. While several strategies exist, in this work the High Throughput Molecular Dynamics (HTMD) framework developed by Doerr *et al* was used due to its availability and ease of use[8].

In the HTMD framework, an initial set of n_{max} short simulations are run. This initial set of simulations is analysed, and new simulations are spawned based on this analysis. Each set of simulations is called an *epoch*, and after each epoch, all the data produced so far is analysed again to inform the next, as depicted in Figure 7.4. The HTMD framework is specifically designed to be modular, so the details of the analysis may be configured depending on the application. The default method uses an MSM model to select conformations in what is called the $\frac{1}{M_c}$ method. In this method, after each epoch, all the data is processed to produce an MSM model with n macrostates, with z_i observations of each macrostate for $i \in [1, n]$. Conformations are selected for re-spawning with probability inversely proportional to z_i — the macrostates less frequently sampled are more likely to be re-spawned.

With this strategy, the process of selecting conformations from which to run additional molecular dynamics simulations is automated. Since novel conformations are favoured, the process will accelerate the exploration of the system. The process is re-

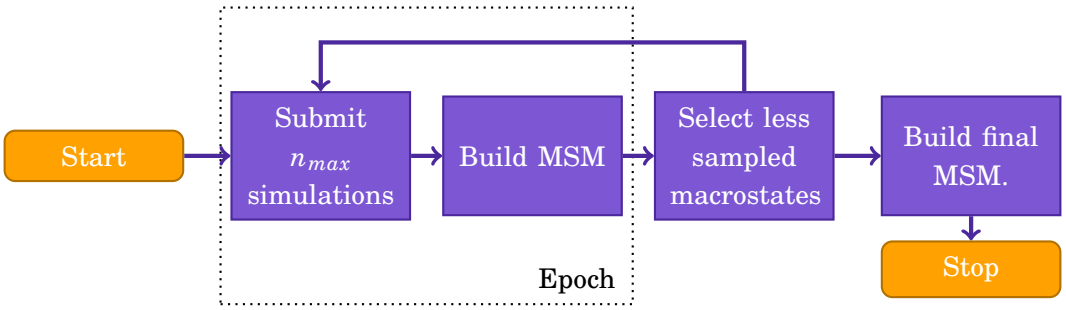


Figure 7.4: Flowchart of the HTMD Adaptive Sampling Procedure.

peated for a user-specified number of epochs n_{epochs}, upon which the final Markov state model can be used to assess convergence. This strategy is a compelling companion to IMD, which can provide a set of potentially relevant configurations.

An initial foray into adaptive sampling using HTMD was undertaken, using six configurations along the proposed slipknotting pathway IMD sampling as the initial conditions, shown in Figure 7.5. An extension to HTMD was written to enable adaptive sampling using OpenMM as the molecular dynamics engine[2]. These initial conditions were equilibrated for two nanoseconds with OpenMM using the Amber10 forcefield in Amber10 GBSA-OBC implicit solvent, with a temperature of 300K maintained using a Langevin thermostat with a friction coefficient of 1 ps^{-1} and a time step of 1 femtosecond. A non-bonded cut-off distance of 2 nm was used, with the length of all bonds that involve a hydrogen atom constrained. Each epoch consisted of a maximum of 8 simulations, each run on a single local GPU. These production simulations were 50 nanoseconds long using the same parameters. After each epoch, a Markov state model was produced as follows. The trajectory data was projected onto a lower dimensionality representation using the distances between all pairs of alpha-carbon atoms, and TICA was used to reduce dimensionality further. The default parameters of 3 TICA dimensions and a TICA lag of 20 frames were used. The projected data were then clustered using the K-Center method using the default heuristic in HTMD for determining the number of clusters to use. From this a Markov state model was generated, again using the default parameters of a lag time of 1 frame. HTMD uses PyEmma to construct the MSM, which uses the largest connected set of states at a given lag time to produce a model, and then uses PCCA to coarse grain the model into macrostates[68]. The default value of 8 macrostates was used, but the default algorithm may reduce this number of macrostates if PCCA produces empty states. Default parameters were used as they are

[2]Available at https://github.com/mikeoconnor0308/HTMD-Adaptive-OpenMM.

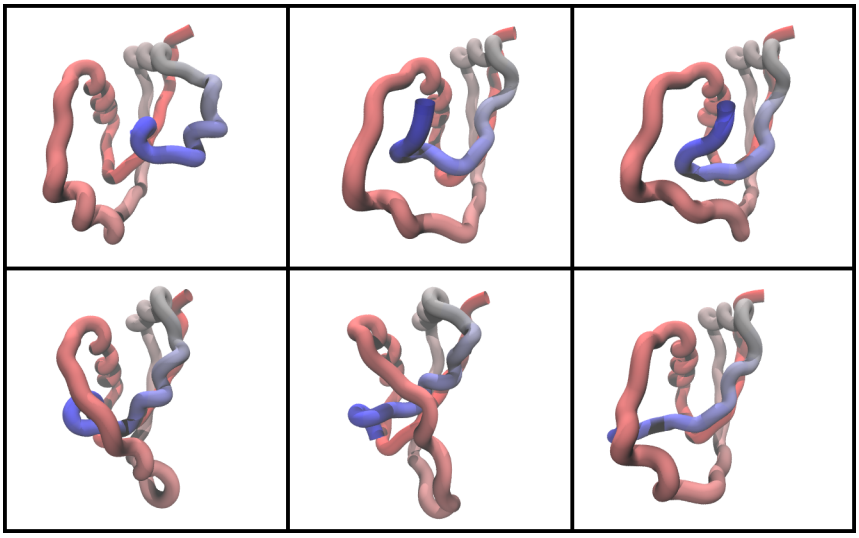


Figure 7.5: The slipknot pathway configurations used to seed the HTMD adaptive sampling run.

designed for use with protein folding simulations[8], and with a lack of prior knowledge about the system, the goal at this point in the process is to accelerate sampling, not to necessarily produce the best Markov model.

Twenty-seven epochs were run with this methodology, totalling 4.1 microseconds of molecular dynamics trajectories. At this point, the sampling was halted to review its progress. Figure 7.6 shows the genealogical tree of trajectories performed with adaptive sampling, providing insight into how the various initial configurations were used by the algorithm to perform sampling. The nodes of the figure are coloured from light green to dark blue as a function of the epoch, with darker shades indicating later samples. As can be seen, the majority of sampling occurs in a tree of trajectories spawned from a single configuration (e1s7), in this case, a knotted one. Rather than using the different configurations to perform sampling in novel conformations, the adaptive algorithm samples heavily from the local region around a particular configuration.

Upon reflection, it is unsurprising that sampling chiefly occurs from a single region of configuration space, if we consider how the Markov state model is generated. The model is constructed from the largest connected set of microstates calculated after dimensionality reduction and clustering, using a maximum likelihood estimation. Upon coarse-graining, the macrostates of the model are then based upon this set of microstates. The fact that it is a maximum likelihood estimation means that it may occasionally determine a different connected set of microstates to be the best fit (this can be seen by the set of simulations spawned from *e1s6* later into the process). However, as

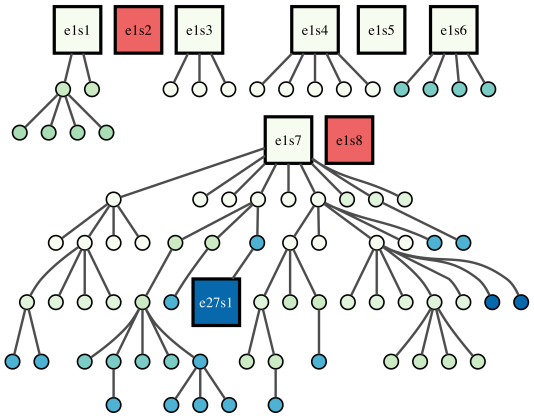


Figure 7.6: Genealogical tree of the simulations run in adaptive sampling. Starting from the large labelled white squares representing the initial configurations, each node represents a simulation, coloured by the epoch from white to green to dark blue. An edge from a node proceeding down the figure represents the spawning on a new simulation from a frame in the preceding simulation. Red squares indicate simulations that failed to run, due to either unstable configurations or node failure. The final epoch's simulations are labelled as dark blue squares to indicate the state of the adaptive sampling algorithm at the end of the run.

a particular set of connected states becomes better sampled, a feedback loop is formed in which that set of states is repeatedly sampled. This means that, by design, the disparate states between which there have not been any transitions will not be sampled further.

One solution to this problem would be to produce a much more fine-grained set of initial seeds, in the hope that they are more likely to result in transitions. However, such an approach would not be guaranteed to succeed, as initial conditions may rapidly fall into minima without transitions between them being observed.

The configurations generated through IMD effectively represent a prior estimation of potential metastable states and transition states between them that can be leveraged to accelerate sampling. The adaptive algorithm needs to include our assumption of connectivity between the states *a priori* until such transitions are observed. One method to facilitate this is to add transitions between each microstate artificially, by manipulating the count matrix clusters $\mathbf{C}$ to indicate that between every pair of clusters a transition has been observed. This results in a fully connected set of microstates, and the resulting macrostates will represent all the configurations sampled thus far.

Adaptive sampling was repeated with this modification made. Additionally, the number of initial configurations was increased to 10, as shown in Figure 7.7. These configu-

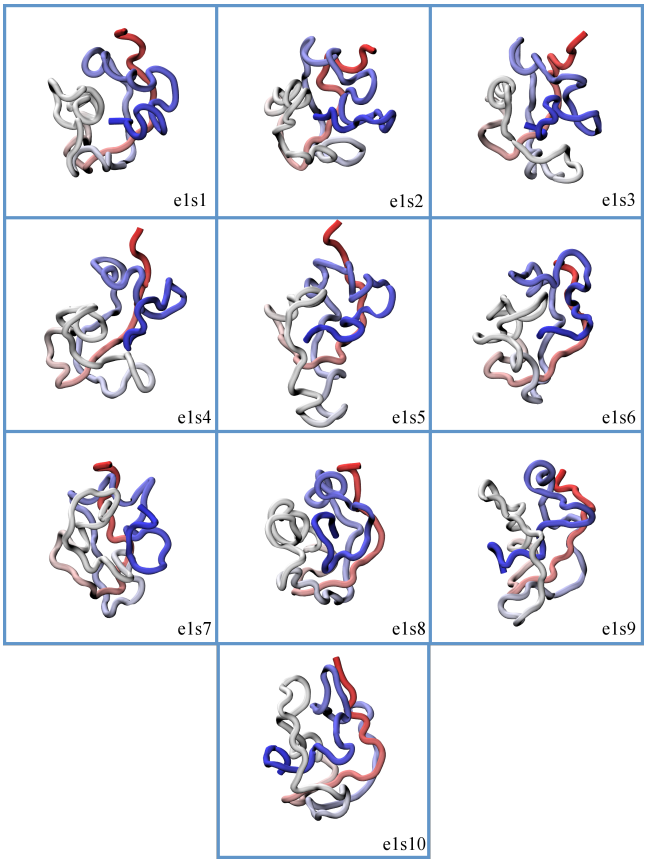


Figure 7.7: Initial configurations used for adaptive sampling. Each tile is labelled with its corresponding simulation identifier.

rations were selected to be various states before and leading up to the slipknot event, as well as one configuration in a loosely knotted configuration. These configurations were selected under the hypothesis that they would be accessible states from one another within simulation timescales. The number of simulations in each epoch was increased to 10 for this run due to increased computational resources becoming available. A total of 14.3 microseconds of molecular dynamics was run, over 62 epochs. With each simulation running on an NVIDIA GTX 1080Ti capable of 500 nanoseconds per day, on a machine with three such GPUs, this represents approximately nine days of computer time.

Figure 7.8 shows the genealogical tree from the adaptive sampling. This time, sampling is slightly more uniform across the seeds, as indicated by the larger spread of samples from different starting configurations as well as the large spread of blue dots indicating that late into sampling different configurations are being explored, but the seeds *e1s1* and *e1s10* constitute most of the sampling. A feedback loop occurs as config-

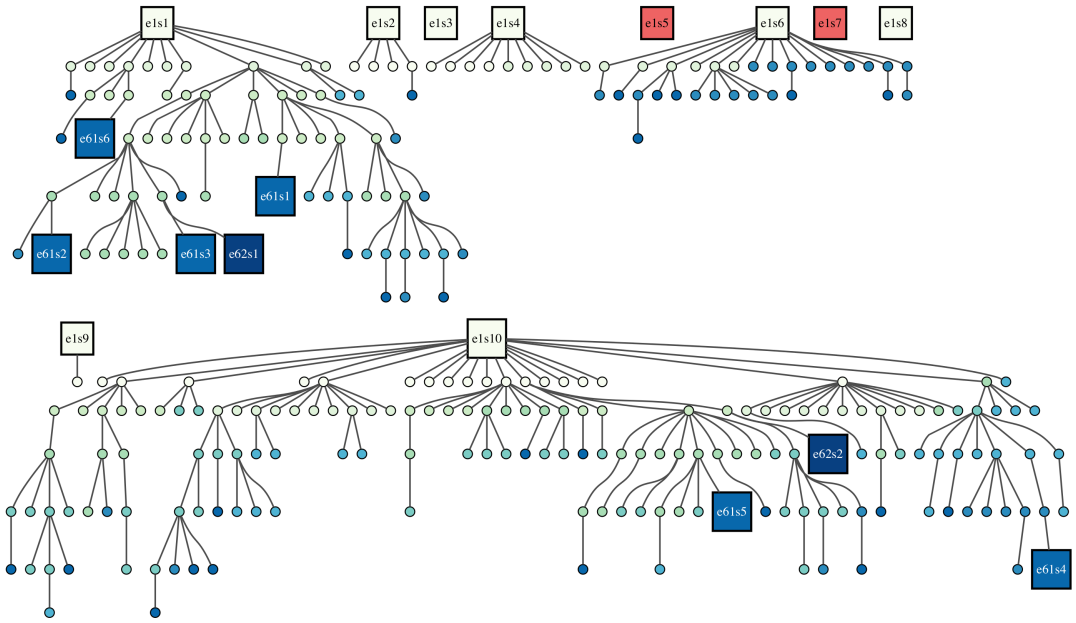
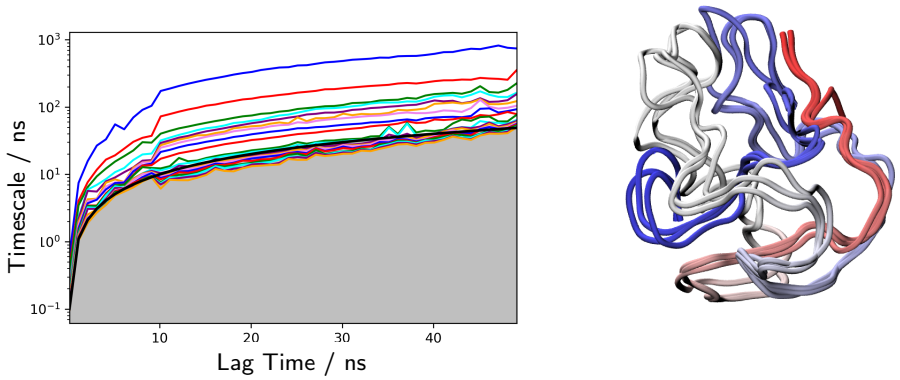


Figure 7.8: Genealogical tree of the simulations run in the second batch of adaptive sampling, with artificial transitions added to the count matrix. Starting from the large labelled white squares representing the initial configurations, each node represents a simulation, coloured by the epoch from white to green to dark blue. An edge from a node proceeding down the figure represents the spawning on a new simulation from a frame in the preceding simulation. The final epoch's simulations are labelled as dark blue squares to indicate the state of the adaptive sampling algorithm at the end of the run.

urations from these seeds are sampled that leads to more macrostates being associated with them.

After adaptive sampling, a Markov state model was constructed to analyse the results. As this model was for analysis of the trajectories, the count matrix was not manipulated. The model was constructed using the same dimensionality reduction methods described above, but the number of macrostates and lag time was determined through inspection of the implied timescales of the model as shown in Figure 7.9. This timescale plot indicates that additional sampling is required because for no lag time are the implied timescales constant[54]. For visualisation purposes, a lag time of 25 nanoseconds was selected to build a model, coarse-grained into three macrostates. These macrostates are visualised in Figure 7.9, and it can be seen that all three macrostates correspond to fluctuations around the loosely knotted state. As the largest connected set of microstates is used to construct the model, we can conclude that, under the discretisa-



(a) (b)

Figure 7.9: a) Implied timescale plot of Markov state model constructed from trajectories in adaptive sampling. b) Sampled conformation from the three macrostates produced by construction of a Markov state model with lag time 25 nanoseconds.

tion method used, there were no transitions between the knotted and unknotted states. Indeed, Figure 7.10 shows the genealogical tree again, this time with samples of the three macrostates highlighted to indicate the trajectory they belong to. There are no highlighted trajectories from any of the nodes that are not spawned from *e1s10*, which further indicates that there were no significant transitions.

With 14 microseconds of molecular dynamics data across a large configuration space, it is possible that there were some transitions that have not yet been detected in this preliminary analysis. However, one can conclude that the large conformational changes associated with knotting did not occur, because one of the most general discretisation approaches, the contact distance between heavy atoms, was unable to detect any such transitions, which would surely have produced disparate microstates had they occurred.

The adaptive sampling strategy did not produce converged sampling of knotting pathways. However, given the size and complexity of this system, one cannot conclude at this point that the combined strategy of IMD with adaptive sampling with HTMD would not be an appropriate method for other systems. In the following section, the approach developed further to perform sampling of loop motions in Cyclophilin A.

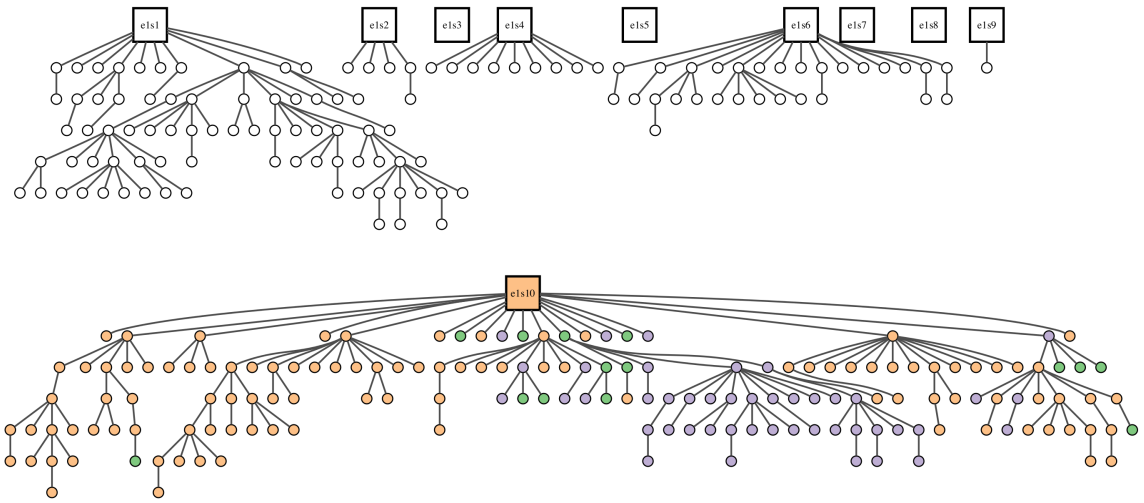


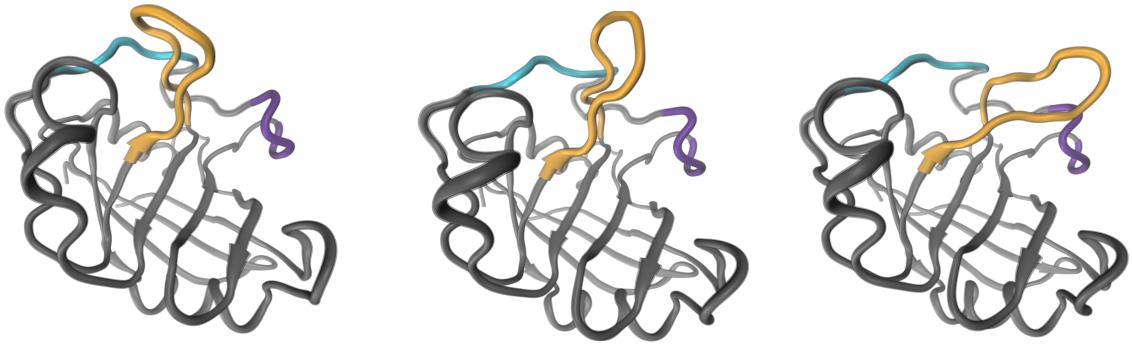
Figure 7.10: Genealogical tree of the simulations run in the second batch of adaptive sampling. The trajectories from which 100 samples of each of the macrostates in the Markov state model originate are indicated by the colours green, purple, orange, while trajectories that do not appear in this sampling are shown in white.

7.2 Accelerated Sampling of Loop Motions in Cyclophilin A

The dynamic nature of protein structures is increasingly being considered in the context of enzyme catalysis. The well-studied protein cyclophilin A (CypA) provides evidence that large-scale collective motions take place[244, 245]. The loop formed by residues Asp66-Gly75 (70s motion, the purple loop in Figure 7.11) has been observed in both experimental and theoretical studies[123, 245] to open on the millisecond timescale. Additionally, the loop formed by residues Ala101-Gln111 (100s motion, see Figure 7.11) has recently been observed to open in molecular dynamics trajectories, occurring on the nanosecond timescale[123]. Whether motions on long timescales have a role during catalysis is an ongoing debate[246], with molecular simulations indicating that it is motions on shorter timescales near the active site that affect catalysis[88].

If the iMD-VR framework is to be used in the study of drug binding and catalysis, it will be necessary to able to characterise any structural changes in enzymes that may be necessary for binding. The applicability of the iMD-VR framework was tested by performing the aforementioned loop motions in cyclophilin A.

Cyclophilin A was simulated in OpenMM, using the Amber FF14SB-ILDN force-field, using particle mesh Ewald summation with a cut-off of 1 nm, with an explicit



(a) (b) (c)

Figure 7.11: Configurations of 100s loop motion in CypA generated in iMD-VR. The loop formed by residues Met100-Ser110 is highlighted in orange, the loop formed by residues Gly65-Gly75 is highlighted in purple and residues Gly80-Leu90 are highlighted in cyan.

solvent of 8610 explicit water molecules modelled with TIP3P. The input structure was based on PDB entry 1AK4[247], provided by Jordi Juarez Jimenez at the University of Edinburgh[123]. The system was simulated with a Langevin integrator at 300K with a friction coefficient of 1 ps^{-1}. A time step of 0.5 fs was used to give the user more control over the manipulation, as it is easier to adjust the bias in response to the dynamics when it is running more slowly. After an equilibration period of one nanosecond, the PLUMED plug-in with IMD enabled was used to connect to the iMD-VR framework.

Both loop motions were explored in iMD-VR using the range of controls described in Chapter 4.2[3]. Qualitatively, loop motions were intuitive to guide with the tools developed, with the notable exception of the build-up of momentum. At the time that this study was performed, the velocity re-initialisation method developed for removing the build-up of momentum described in Chapter 4.2 had not been implemented, and so care had to be taken not to introduce too much energy into the loop motions. Indeed, it was this study that motivated the development of the velocity re-initialisation method. While both loop motions were explored in iMD-VR, in what follows attention is focussed on the 100s loop motion - since it has previously been observed to occur on short timescales, it ought to be an easier target for follow-up simulation.

Some representative configurations generated with iMD-VR are shown in Figure 7.11, including the native state, in which the 100s loop (highlighted in orange) is in contact with residues Gly80-Leu90, and two states in which the loop has been moved away from this starting configuration towards the 70s loop.

[3]A video of the 70s loop motion is provided at `https://vimeo.com/306778545`.

Three trajectories were generated with iMD-VR, each starting from the native state. In previous applications, interesting configurations have been chosen by hand. Here, some more automated methods are proposed for selecting configurations appropriate for seeding follow-up sampling with Markov modelling.

7.2.1 Analysis of iMD-VR Trajectories

In the MJ0366 application, structures for follow-up sampling were selected by hand. In this section, an exploratory analysis is performed to identify methods for automatically extracting representative structures from iMD-VR trajectories of the cyclophilin loop motion.

The first requirement in processing iMD-VR trajectories is the determination of what happened in an iMD-VR trajectory and the pruning of any outliers that are not useful for further analysis. For this application, the fraction of native contacts[248] is used, which provides a measure of how close to the native state a given configuration is. Native contacts are defined here as a distance between heavy atoms i and j in the native state of less than 4.5 Å, subject to the constraint that the residues are at least four residues apart from one another in the protein sequence. The fraction of native contacts, $Q(X)$, which measures how many of the native contacts are present in a given configuration X, is defined as

$$Q(X) = \frac{1}{N} \sum_{(i,j)} \frac{1}{1 + \exp\left(\beta \left[r_{ij}(X) - \lambda r_{ij}^0 \right]\right)},$$

where the sum runs over the N native contacts, (i,j), $r_{ij}(X)$ is the distance between the pair i and j in the configuration X, r_{ij}^0 is the distance between the pair in the native state, β is a smoothing parameter taken to be 5 Å^{-1} and λ accounts for small fluctuations during contact formation, taken to be 1.8 for an all-atom model.

Figure 7.12 shows the native contact fraction for each iMD-VR trajectory. From this, it can be seen that two of the trajectories make excursions away from the native state before returning towards it, while the trajectory coloured in orange trends away, fairly drastically, from the native state. Manual inspection of this trajectory shows a movement of the 100s loop towards residues Gly65-Gly75, but upon returning too much momentum was introduced into the loop, resulting in severe distortion. The trajectory in green is particularly noteworthy as it shows that the user can return the loop to a configuration that is very close to the native state, with a maximum proportion of native contacts of 0.996 restored. This is a very encouraging result, as it provides evidence

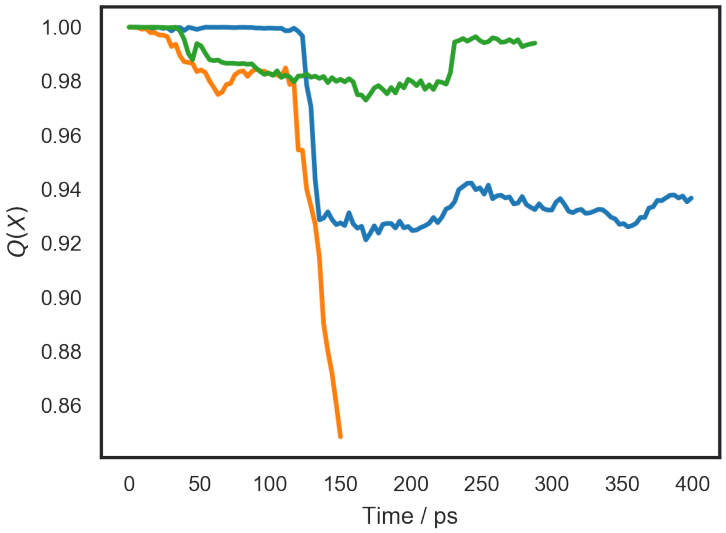


Figure 7.12: Fraction of native contacts, $Q(X)$, over the course of three IMD trajectories in which the 100s loop motion of CypA was explored.

that it is possible, if the user is skilled and careful, to perform subtle, reversible reconfigurations of the protein structure with the iMD-VR framework. The blue trajectory represents an exaggerated loop motion (as seen in Figure 7.11c), but the user manages to return the loop towards the native state.

The native contact fraction metric was used to remove undesired configurations, by removing frames with a value of $Q(X)$ of less than 0.925. This value was chosen from inspection of Figure 7.12, and is clearly context dependent.

7.2.2 Dimensionality Reduction and Feature Extraction

With a set of potentially interesting configurations, a method was sought to choose a representative subset of configurations for follow-up sampling automatically. In what follows, the conformations are clustered in a lower-dimensionality representation and sampled uniformly from the clusters to form the subset with which to carry out additional sampling.

A commonly used method to reduce dimensionality is principal component analysis (PCA)[126, 249]. Given a matrix of feature values, PCA produces a linear transform onto a q-dimensional matrix via a transformation matrix $\mathbf{A}_q \in \mathbb{R}^{N\times q}$, where N is the number of dimensions of the original dataset, that best encompasses the variance in the data. Let $\mathbf{X} \in \mathbb{R}^{m\times N}$ be the matrix of m samples of some N dimensional data, adjusted such

that the mean of each column has a mean of zero. Constructing $\mathbf{A}_q$ to have columns consisting of the eigenvectors corresponding to the q largest eigenvalues of the covariance matrix $\mathbf{X}^T\mathbf{X}$ yields the projection that preserves the largest amount of variance. The q-dimensional projection can then be calculated as

$$\mathbf{T} = \mathbf{X}\mathbf{A}_q,$$

where $\mathbf{T} \in \mathbb{R}^{m \times q}$ is the dataset projected onto the q principal components.

What features should we use to represent our molecular dynamics trajectory? Without any prior knowledge, a reasonable feature set in this application is the distances between all pairs of heavy atoms. PCA was applied to this feature set, using the implementation provided by the python package scikit-learn[60]. Projection onto the first two principal components, which in this case explain 40.4% and 22.7% of the variance of the data respectively, results in the reduced dimensionality representation of each trajectory shown in Figure 7.13. The first observation to make here is that each trajectory is distinguishable in the lower dimensionality representation. This demonstrates the flexibility and size of the configuration space in protein dynamics that can be explored with interactive molecular dynamics, even in a simple loop motion. The overlapping region between all three trajectories represents the loop motion explored by all three trajectories, with excursions from the native state at around (-8,0) in PCA space, with distinct regions explored by all three trajectories as slightly different loop motions were produced. The distinct region near (-10, -2) in PCA space in the orange trajectory (panel D) shows the trajectory beginning to distort the protein.

One could immediately use this reduced representation to generate a spread of configurations for the follow-up sampling, by clustering the reduced dimensionality into the desired number of configurations and picking one from each cluster. However, it can be desirable to be able to understand which features, i.e. which inter-residue distances, are important in these trajectories. Such features are more intuitive and can be used with many rare event methods.

This is the remit of *feature selection*, which is distinct from dimensionality reduction in that rather than simply reducing the dimensionality of the data (in whatever projection best reduces it), we seek to extract a few key features from the original set that best explain the data. Many methods for this are applicable in different contexts, many of which are available in the python package scikit-learn[60]. Since PCA is a common dimensionality reduction technique commonly used in the analysis of molecular simulations, the use of Principle Feature Analysis[249] was investigated as a feature selection

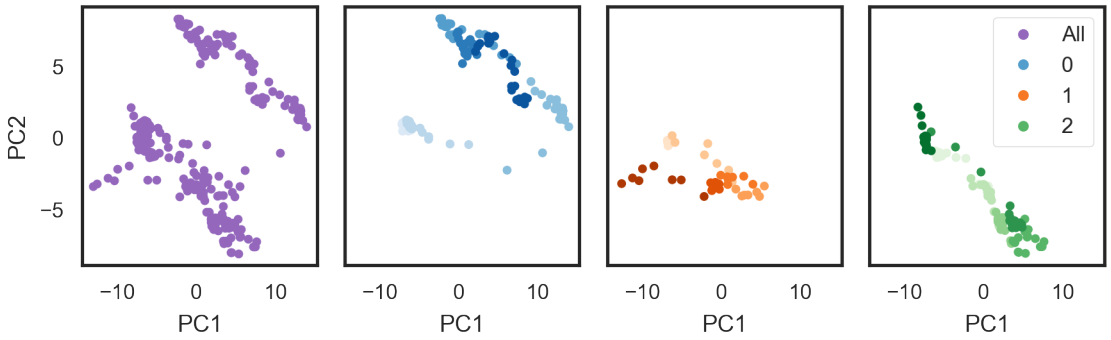


Figure 7.13: Trajectories from IMD projected onto the first two principal components of PCA using the all heavy-atom contact distances as the features. Panel A shows all trajectories projected on to the first two PCA components, while panels B, C, and D show each trajectory separately. Trajectories are coloured from light to dark shades to indicate the passage of time.

method, which uses PCA and then attempts to back out the features that best map onto the reduced dimensionality. It does this by observing that each row of the PCA matrix $\mathbf{A}_q$ represents the projection of a particular feature onto the principal components. Features that are highly correlated will have row vectors associated with them in $\mathbf{A}_q$ that share similar absolute weights in each component. This property is used to select features by clustering the rows of $\mathbf{A}_q$ into a number of clusters equal to the desired number of features. The features corresponding to the rows of $\mathbf{A}_q$ closest to each cluster centre are then extracted. The rationale here is that the feature that is closest to the cluster centre best represents that particular region of the reduced dimensionality space and all the features belonging to the same cluster can be considered closely correlated.

Applying the method to the dataset of all contact distances produces unsatisfying results. With three features set for extraction, the features extracted are the distances between residues Arg55 and Thr73, residues Val12 and Hie70, and residues Met1 and Val127. The projection of the iMD-VR trajectories onto each pair of these features is shown in Figure 7.14. These features are not directly involved in the 100s loop motion, but adjustments in these distances that are correlated with the loop motion are clear. For example, in the blue trajectory, the distance between residues Val12 and Hie70 increases then decreases throughout the trajectory, as it relaxes in response to the loop motion performed by the user.

This seemingly random set can be understood by viewing the structure of the PCA matrix $\mathbf{A}_q$ and the resulting clustering that is used to select features, as shown in Fig-

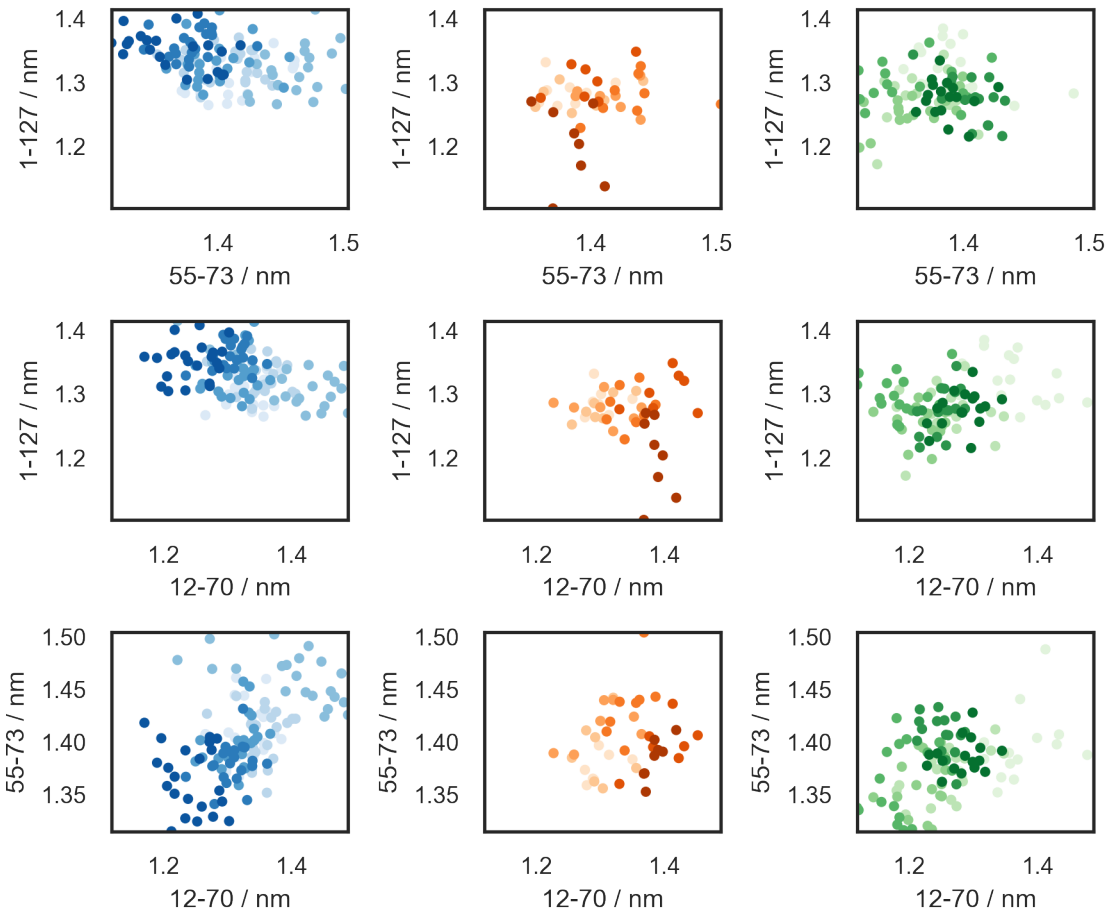


Figure 7.14: The three IMD trajectories projected onto the three pairs of features extracted via PFA using all heavy atom distances as features. These features are the contact distances between residues Arg55 and Thr73, residues Val12 and Hie70, and residues Met1 and Val127. Each panel shows a different pair of projections. Each trajectory is shown in a different colour, shaded from light to dark to indicate the passage of time.

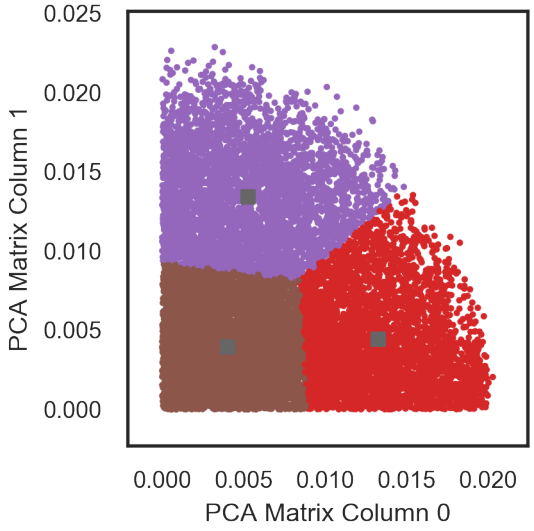


Figure 7.15: A plot of the rows of the PCA matrix $\mathbf{A}_q$ with all heavy-atom contact distances used as features, along with the clusters produced by PFA. Cluster membership is indicated by different colours, with cluster centres indicated by grey squares.

ure 7.15. The absolute values of the rows of the matrix $\mathbf{A}_q$ are plotted, and there are many features with similar weights, indicating that they are correlated. With so many features exhibiting correlation, the nearest feature to a cluster centre merely extracts one of the features that exhibit this correlation. It appears that the PFA method is not effective at extracting features from a very high dimensional feature space in which many features are correlated. This is a challenge for feature selection algorithms in molecular dynamics. Because the system is coupled through interatomic forces, there can be many correlations in the dataset, but it can be difficult to extract the causation, i.e. what features drove the changes in the system.

In this analysis, we have neglected some crucial information from the trajectories produced with iMD-VR: the user has provided guidance into which features are relevant. The selections made by the user and the sets of atoms upon which interactive forces or restraints were applied effectively provide a dimensionality reduction and preliminary feature extraction. In the trajectories, the residues in ranges Lys82-Asn87, Met100-Ser110 and Phe67-Gly75 were selected by the user. Repeating the analysis but this time using the contact distances between heavy atoms in residues contained in the ranges Lys82-Asn87, Met100-Ser110, Phe67-Gly75 as the feature set produces very different results. Figure 7.16 shows the trajectories projected onto the first two PCA components using this reduced set of distances. The variance explained by these two

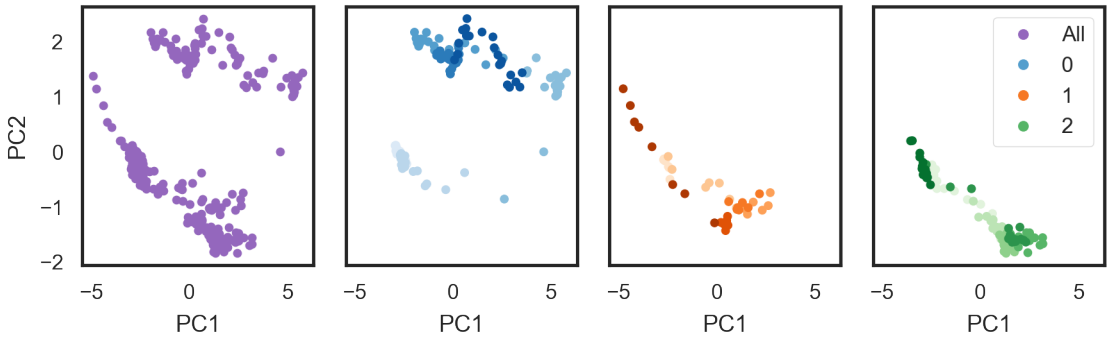


Figure 7.16: Trajectories from iMD-VR projected onto the first two principal components of PCA using the user determined contact distances as the features. Panel A shows all trajectories projected on to the first two PCA components, while panels B, C, and D show each trajectory separately. Trajectories are coloured from light to dark shades to indicate the passage of time.

components is 66% and 18% respectively, indicating that the variance in this reduced set of distances can be explained better by fewer components. The structure is largely comparable to that produced by using all the heavy atom contacts, but there is a notable exception in the orange trajectory (panel C in Figure 7.16 and Figure 7.13). Here, the reduced feature set suggests that the orange trajectory passes back over the native state, while the full set of contact distances indicate that the orange trajectory is heading off into a new area of phase space (the beginning of the distortion of the loop introduced by an excess of energy into the system by the user).

Figure 7.17 shows the PCA $\mathbf{A}_q$ matrix values for the reduced feature space, and the clustering performed to select three features. Because the feature set is reduced significantly, the number of correlated features is reduced, and so there is a clearer structure to the feature set. The features extracted in this case were the distances between residue pairs Gly74 and Ala103, Lys82 and Gly104, and Asp85 and Gly72.

These extracted features make intuitive sense when the trajectories are projected onto each pair of these distances as shown in Figure 7.18. The top row of this figure is the most revealing. The increasing distance between residues Lys82 and Gly104 corresponds to the loop moving away from the native state (the orange loop moving away from the cyan region in Figure 7.11), while the distance between residues Gly74 and Ala103 corresponds to the loop moving towards the loop formed by residues Gly65-Gly75 (Figure 7.11c). The transition from light to dark shades of the colours shows the user attempting to bring the loop back to its native position. The difference between the dis-

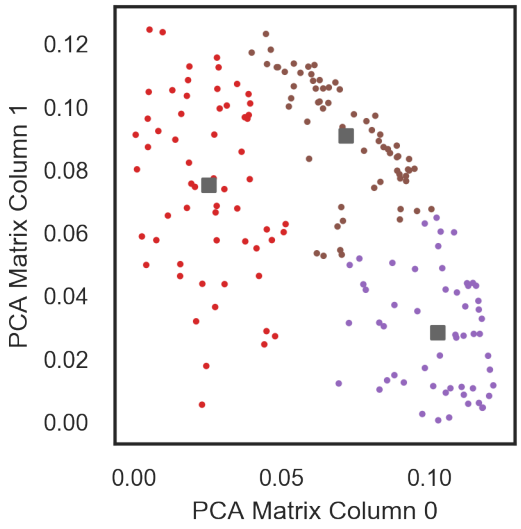


Figure 7.17: A plot of the PCA matrix $\mathbf{A}_q$ with user-determined contact distances used as features, along with the clusters produced by PFA. Cluster membership is indicated by different colours, with cluster centres indicated by grey squares.

tances between residues Lys82 and Gly104 at the end of the blue and green trajectories respectively clearly indicates the greater success in this attempt in the green trajectory. Also visible is the greater extension of the loop motion achieved by the blue trajectory, compared to the relatively modest motion in the green trajectory. The distortion in the protein loop at the end of the orange trajectory can be seen in the increase in distance between residues Gly74 and Ala103.

While the distance between residues Asp85 and Gly72 does vary throughout each trajectory, it is not particularly correlated with the other features. For example, in the blue trajectory the distance between residues Asp85 and Gly72 increases as the distance between Lys82 and Gly104 increases, while in the green trajectory, the relation is the opposite.

The PFA method provides a method to extract features, especially if one is already using PCA to perform dimensionality reduction. However, it performs poorly if the feature set has high dimensionality, with many correlated features. The fact that it was successful on the reduced feature set is not particularly impressive, as any of the pairs of distances between the various structures selected (the orange, purple and cyan regions in Figure 7.11) produce similar results. This can be seen by changing the number of features selected, which changes the features selected, as the positions of clusters change. For example, increasing the number of features to five results in the residue

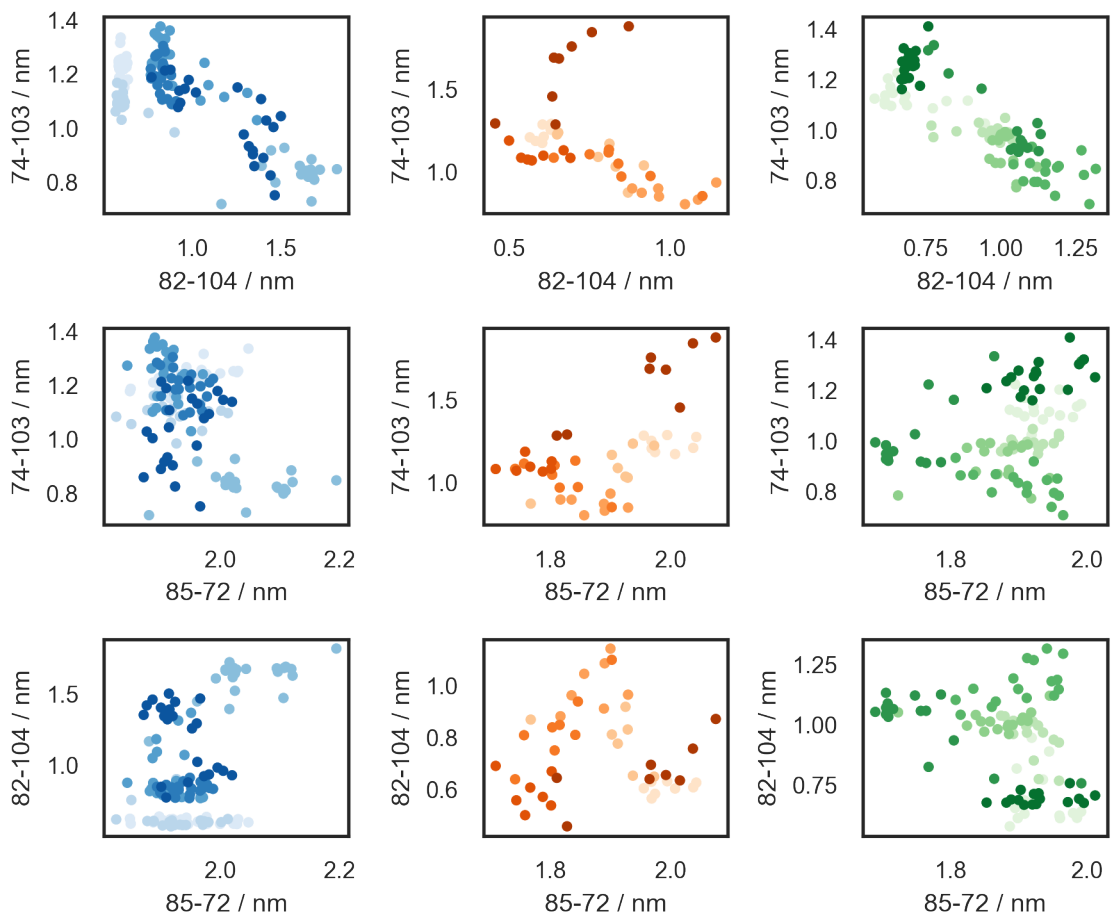


Figure 7.18: The three IMD trajectories projected onto the three pairs of features extracted via PFA. These features are the contact distances between residues Gly74 and Ala103, Lys82 and Gly104, and Asp85 and Asn72. Each panel shows a different pair of projections. Each trajectory is shown in a different colour, shaded from light to dark to indicate the passage of time.

pairs (Asn72, Met100), (Thr68, Thr107), (Glu84, Thr73), (Lys82, Gly109) and (Lys82, Ala103) to be selected. These residue pairs exhibit similar characteristics to those discussed above. It is also not immediately clear how important each feature is, or how many to use.

In this case, the regions selected by the user provides a good initial dimensionality reduction from which the trajectories could be characterised. Further study on different datasets would be required to determine whether this is generally the case. More advanced feature selection methods, such as those provided in scikit-learn should be explored to develop more robust workflows[60].

The three features selected were used to cluster the trajectories into 15 clusters using the KMeans clustering algorithm in scikit-learn. From these clusters, a single configuration was randomly selected. These 15 configurations, shown in Figure 7.19, representing a variety of conformations throughout the 100s loop motion. These initial conditions include a range of configurations including the native state and a variety of open loop positions. These configurations were then used to seed adaptive sampling with HTMD.

7.2.3 Adaptive Sampling with HTMD: Revisited

The adaptive sampling approach using Markov state models was revisited for this application, the reasoning being that since it is a smaller configurational change that is being sampled, compared to knot tying and protein folding, that it may be possible to converge a model.

The approach for adaptive sampling was developed further. In the previous attempt, the transition matrix used by HTMD to construct a Markov model was manipulated to indicate that all the observed microstates should be considered connected. This approach was quite invasive from an implementation standpoint, and furthermore, enforcing a particular structure to the Markov model defeats the purpose of using one for sampling. Instead, let us consider the desired behaviour of a sampling approach. We would like for sampling to take place from each of the initial conditions provided, and for any new conformations discovered to be sampled further. Additionally, we would like for regions of configuration space that have been visited less frequently to be sampled further.

Such sampling can be achieved without the construction of a full Markov model. Before construction of an MSM, one applies dimensionality reduction techniques and clusters the data. This projected dataset provides enough information for automated adaptive sampling.

After applying appropriate projection and dimensionality reduction techniques, the data is clustered into M clusters, where clusters of less than n members (a user-set parameter) are merged. The initial conditions for the required number of simulations for the next epoch, n_{sims}, is then selected by sampling from the clusters, such that less visited clusters have a higher chance of being selected. This is similar to the $1/M_c$ method used by HTMD, except applied to clusters. The proportion of frames in each cluster, p_c, out of the total number of frames, N, is inverted and normalised to give

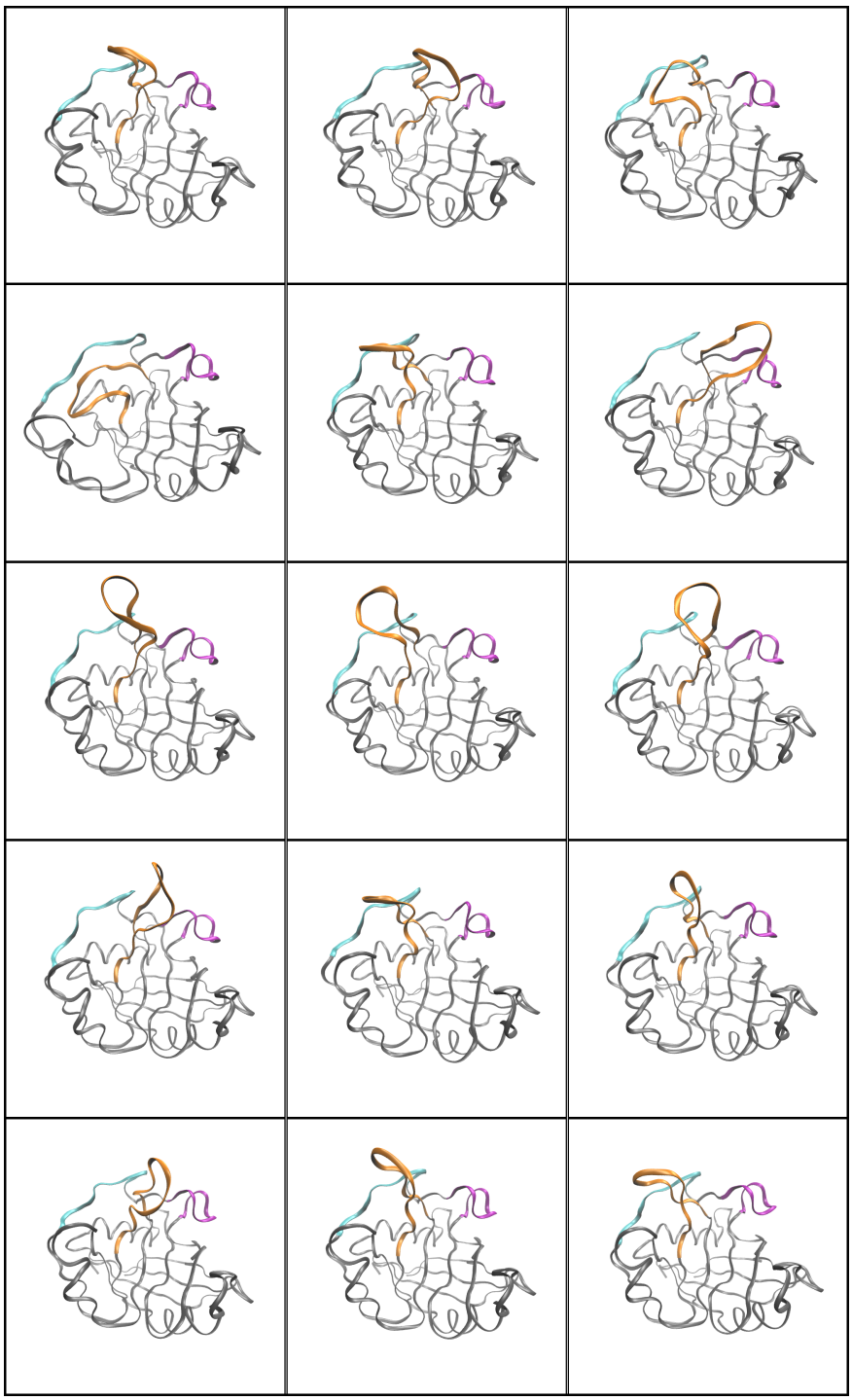


Figure 7.19: The cyclophilin A configurations generated with iMD-VR extracted and used for follow-up sampling. The loop formed by residues Met100-Ser110 is highlighted in orange, the loop formed by residues Gly65-Gly75 is highlighted in purple and residues Gly80-Leu90 are highlighted in cyan.

a probability of being sampled, p_{spawn} that is inversely proportional to the frequency with which the cluster has been sampled:

$$p_{spawn} = \frac{1/p_c}{\sum_c 1/p_c},$$

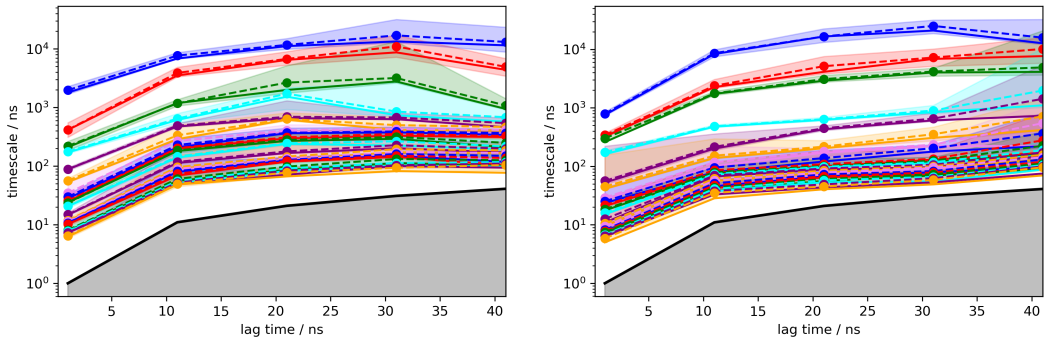
where

$$p_c = N_c/N.$$

Here, N_c is the number of frames in cluster c. As noted by Doerr *et al.* in their description of the adaptive sampling algorithm used by HTMD[242], using cluster labels (or microstates) as a basis for choosing samples can introduce statistical error due to mislabelling of clusters. However, the advantage of this method is in being able to automatically sample disparate configurations between which transitions have not yet been observed, while the procedure used by HTMD will only sample from a connected set of macrostates.

As mentioned above, the HTMD framework is modular, and so this new sampling approach could be implemented within it, allowing for the rest of HTMD's functionality in managing the adaptive sampling to be used. This modified procedure was used to perform additional sampling of the 100s loop motion, using the initial conditions generated with interactive molecular dynamics. Fifteen initial conditions with various configurations of the loop were used as the initial seeds, with 25 simulations of 50 ns each per epoch. The distances between the α-carbon atoms in residues Gly80 to Ile89, Met100 to Ser110, and Phe67 to Gly75 were used as the projection for clustering, drawn from the selections made by the user during the iMD-VR sessions, and TICA was used for dimensionality reduction with the default number of dimensions of 3. Clusters of less than ten members were merged. The simulations were run with OpenMM using the Amber03 forcefield with the accompanying GBSA-OBC implicit solvent (for computational efficiency) at a target temperature of 300K with a time step of 2 fs using a Langevin integrator. A cut-off of 2 nm was used for non-bonded interactions, and any bonds involving a hydrogen atom were constrained to a fixed length. The simulations were halted after 19 epochs, totalling 20.1 μs of molecular dynamics data.

The same approach was used with vanilla HTMD adaptive sampling, using only the native PDB state as an initial generator, to provide a comparison to the iMD-VR seeded sampling.



(a) (b)

Figure 7.20: Implied timescale plots for the Markov models produced with A) adaptive sampling from iMD-VR seeds and B) adaptive sampling from the native state only. Produced with PyEmma using Bayesian sampling of the posterior to compute uncertainties[68].

7.2.4 Results

After the sampling, a Markov model was constructed from all of the trajectories. A separate model was produced for the sampling seeded from VR trajectories and for the sampling seeded from the native state. The model used the contact distances between the alpha carbon atoms in residues Phe60 through Glu120. The set of residues to include in the final sample was expanded to capture any additional motions not described by the residues Gly80 to Ile89, Met100 to Ser110 and Phe67 to Gly75. The default K-Centers clustering algorithm in HTMD was used, and TICA was applied with a lag time of 20 frames, using two dimensions.

Figure 7.20 shows the implied timescales for each model, from which we can conclude that neither model is particularly well-converged, as the timescales are not constant for any value of lag time, with overlapping error bars in different implied time scales. Nevertheless, for the purpose of analysing the performance of the sampling algorithm, Markov models using the lag time 25 ns were constructed for each dataset.

This set of parameters produced a Markov model that used 67.7% of the data for the dataset produced with trajectories seeded from VR, while 93% of the data was used in the corresponding Markov model produced from the native state only.

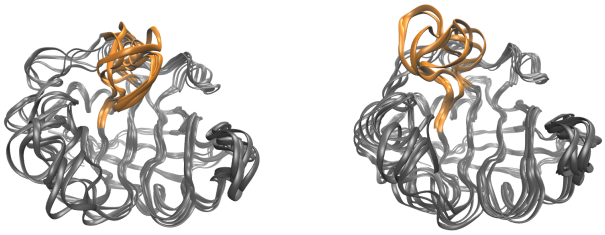
The model seeded from VR was coarse-grained with PCCA into four macrostates for visualisation purposes, and five random samples of each macrostate are shown in the panels A through D of Figure 7.21, along with their observed populations. These samples indicate that a range of small motions in cyclophilin A were observed, but the

states sampled do not reproduce all the configurations sampled with interactive molecular dynamics (see Figure 7.19). Panel E of Figure 7.21 shows the macrostates produced from the adaptive sampling from the native state only. Here, a much smaller range of fluctuations in the loop is observed. From this, we can conclude that the trajectories seeded from VR did sample a broader range of configuration space compared to adaptive sampling from the native state only. However, the fact that 99% of the population is labelled macrostate three, and that specific open/closed states are not recovered with significant populations, such as those found in previous studies[123], suggest either poorly converged sampling, a poor discretisation, or that the use of GBSA-OBC implicit solvent model has significantly impacted the interactions of the system. Additionally, there were no observations of the 70s loop opening.

Figure 7.22 shows the genealogical tree of adaptive sampling seeded from VR, with simulations contributing to a macrostate highlighted. The spread of the tree indicates that sampling was more uniform across initial conditions than the previous approach. However, there is still a tendency to favour spawning trajectories from some trees rather than others. This is due to a combination of selection bias and the relatively small number of spawning opportunities. By default, in HTMD the number of clusters generated increases as a function of the number of simulation frames. As a given 'tree' is sampled more, it generates new, slightly different conformations, and is thus is more likely to have samples included in a greater proportion of both new and existing clusters, and thus more likely to be chosen for resampling. Additionally, the merging of small clusters further decreases the likelihood of very rarely visited conformations being sampled.

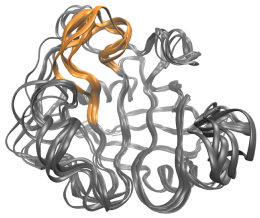
This can be seen in Figure 7.23, which shows the cluster membership proportions and the resulting cluster spawn probability after the first five epochs of sampling. While the resulting spawn probabilities do favour less sampled clusters, there are so many clusters that the likelihood of a particular cluster from a particularly poorly sampled tree being chosen for spawning is relatively low. For instance, the least sampled cluster in Figure 7.23 has a probability of being resampled of only 0.007. The Bayesian approach taken by the FAST algorithm [250], which estimates the likelihood that sampling from a given state is likely to lead to the discovery of new states, may provide a better solution to this problem.

While the sampling was less uniform than intended, the existence of different macrostates within and across separate simulation trees indicates that, under the discretisation used, transitions between disparate seed conformations produced with iMD-VR were observed. However, nearly a third of the trajectories are disconnected from the model,

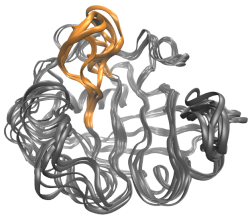


(a) Macro 0 (0.5%)

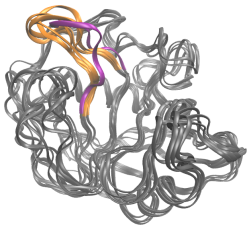
(b) Macro 1 (<0.1%)



(c) Macro 2 (<0.1%)



(d) Macro 3 (99.4%)



(e) Native Sampling

Figure 7.21: Panels A through D: Visualisation of the macrostates of a Markov state model produced from adaptive sampling of Cyclophilin A seeded from trajectories produced with molecular dynamics in VR, along with their populations. Each panel shows five samples of the given macrostate, sampled uniformly at random. Panel E: The macrostates produced through adaptive sampling from the native state. The loop formed by residues Met100 to Ser110 is highlighted in orange. In panel E, the native state loop position is highlighted in purple. Visualisations produced with VMD[193].

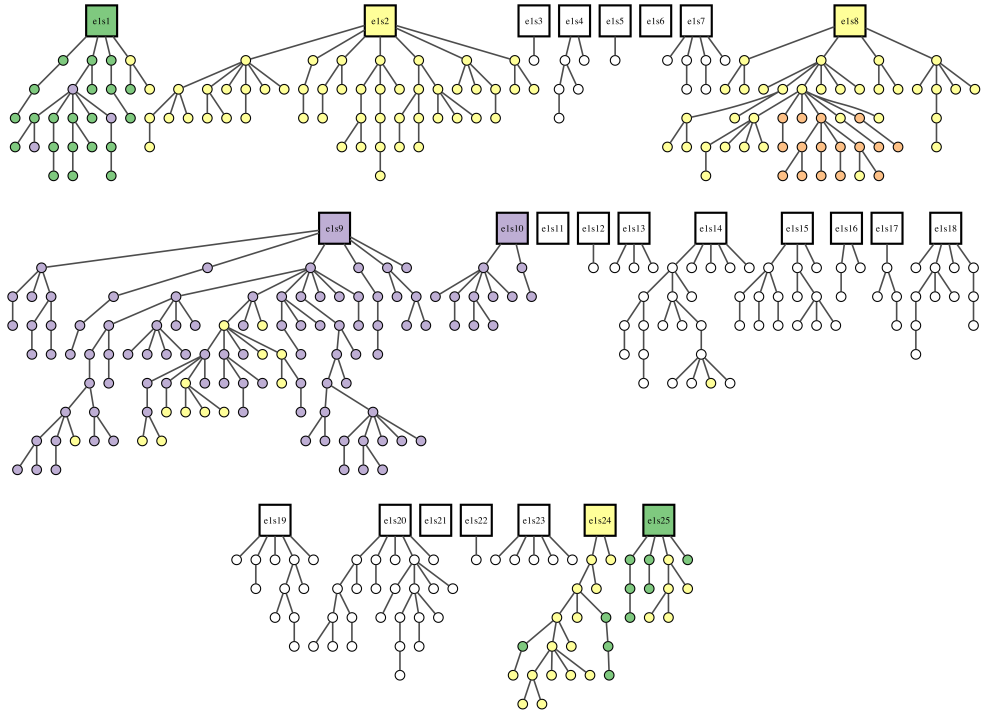


Figure 7.22: Genealogical tree of the simulations run in the adaptive sampling of CypA loop motions. The trajectories from which 100 samples of each of the macrostates in the Markov state model originate are indicated by the colours green, purple, orange and yellow, while trajectories that do not appear in this sampling are shown in white.

indicating transitions were not observed between all the configurations seeded from VR.

With a poor Markov model that is not well converged, it is desirable to understand further the region of configuration space sampled. To achieve this, the data was projected, at a stride of 5 ns, onto the residue distances extracted during the original analysis of the virtual reality trajectories. While these features certainly do not capture all of the relevant dynamics of the system, they provide some intuitive insight. A heat map of the sampled values is shown for each pair of distances in Figure 7.24, with the 15 initial seeds projected on top in orange. It is clear from this projection that sampling was focussed on configurations characterised by a shorter distance between residues Lys82 and Gly104, with the extensions which would be indicative of the larger loop motions produced in the iMD-VR sessions sampled less frequently, if at all. This can be seen in the macrostates extracted in Figure 7.22, compared to the initial conditions in Figure 7.19. Additionally, the sampling explored a range of configurations characterised by a

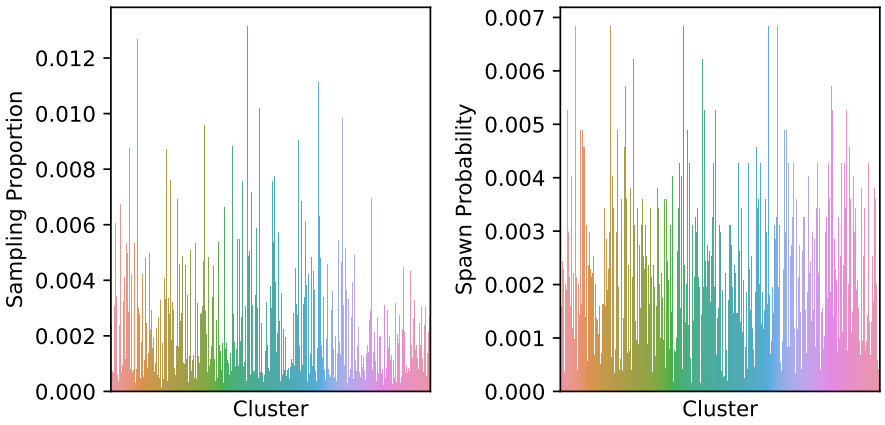


Figure 7.23: The cluster proportions (left) and resulting spawn probabilities (right) after the first five epochs of adaptive sampling of the loop motion in cyclophilin A.

short distance of less than 0.5 nm between residues Lys82 and Gly104 that were not sampled in iMD-VR.

This sampling is reflected in the macrostate populations. Macrostate 3, accounting for 99% of the population, is characterised by a series of long lived contacts. In a sample of 100 configurations from macrostate 3, there are 99 occurrences of a contact between Glu81 and Asn106, with an average distance of 0.25 nm, 99 occurrences of a contact between Phe83 and Gly109, with an average distance of 0.26 nm, and 95 occurrences of a contact between Asp85 and Asn102, with an average distance of 0.25 nm. The other (minutely populated) macrostates can be characterised by different contacts forming, such as a contact between Phe83 and Asn108 in macrostate 0, Phe83 and Asn102 in macrostate 1, and Asp85 and Asn102 in macrostate 2. Such long lived contacts in macrostate 3 indicate that the sampling was trapped in this state, with rare transitions to other closely related loop conformations.

A plausible explanation for this behaviour, and the lack of the opening of the 70s loop, is overstabilisation due to the use of the GBSA-OBC solvent. It has previously been shown in simulations of small peptides that GBSA-OBC implicit solvent models lead to overstabilised salt bridges and a tendency for stronger electrostatic effects between residues, when compared to explicit solvent models[251, 252]. The residues in the 100s loop forming long lived contacts are not those associated with the formation of salt bridges, but the dynamics certainly appears to have become trapped with sustained electrostatic interactions between the aforementioned sets of residues. A comparative study with explicit solvent and other implicit solvent models would have to be made to

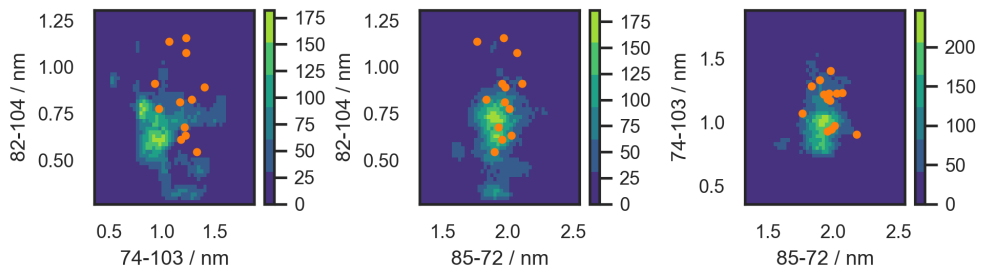


Figure 7.24: The adaptive sampling trajectories, seeded from VR, projected onto the three pairs of features extracted via PFA. These features are the distances between residues Gly74 and Ala103, Lys82 and Gly104, and Asp85 and Asn72. Each panel shows a different pair of projections. A heat map is used to indicate the density of sampling of each projection. The initial conditions of the adaptive sampling trajectories, seeded from VR, is overlaid in orange.

determine whether the force field has caused this behaviour, or if loop closing events were simply unsampled due to lack of computational resources.

From this analysis, it can be concluded that the adaptive sampling seeded from VR trajectories sampled a considerably wider range of configurations than an equivalent sampling strategy from just the native state. However, the trajectories did not produce further samples of several of the configurations produced in VR. Instead, they sampled a range of other, smaller, loop opening motions, and did not recover the equilibrium populations previously reported[123]. Given the flexibility of the configurations involved, this is perhaps not surprising. Indeed, it is expected for the dynamics to relax into a variety of minima. It seems plausible that the choice of forcefield and use of an implicit solvent, while enabling longer timescales to be sampled, may have affected the populations observed. The poor sampling makes it impossible to quantitatively determine whether the virtual reality trajectories were unrealistic, or that they represent a rare event over some larger energy barriers that went unsampled.

7.3 Discussion and Conclusions

In this chapter, an exploration into using adaptive sampling with Markov state models to sample the knotting pathway of MJ0366 and loop motions in cyclophilin A was performed. The iMD-VR framework was used to produce initial configurations which were extracted to form the initial configurations used in adaptive sampling. The resulting trajectories, which can be generated quickly, provide data that can be used to perform

analysis and extract low dimensionality representations, collective variables and initial conditions.

It was also observed, however, that without care, it is easy to introduce too much energy into the system and produce high energy pathways and configurations. Additionally, the strength of the biasing potentials used and the timescales in which manipulations take place means that the resulting pathways are unlikely to follow a minimum energy pathway. Instead, they merely serve as a quick way of generating initial conditions and hypotheses, which must be validated.

Automating the procedure from interactive sampling to feature selection was explored using principal component analysis and principal feature analysis. It was found that PCA was able to distinguish between the characteristics of the different VR trajectories. Principal feature analysis was not successful on inputs with large numbers of correlated features, but reducing the feature set to the distances between residues selected by the user in the iMD-VR sessions proved an effective strategy for reducing the dimensionality to allow feature selection to take place. Incorporating the user's intuition in this manner appears to be a promising way to automate the processing of iMD-VR trajectories for follow-up sampling.

For performing follow-up sampling, the HTMD framework is easy to use and lends itself well to automation. It was observed, however, that the adaptive sampling strategies based on Markov models are primarily designed to start from a single point in configuration space, and so modifications to the sampling algorithm were made to perform automated sampling from multiple starting points of configuration space.

With the computational resources available, the sampling of neither system produced converged statistics. Furthermore, the decision to use implicit solvent models in an attempt to expediate sampling may have dramatically affected the resulting sampled populations. However, the sampling undertaken does provide some insights into the topology of protein energy landscapes, and the use of adaptive sampling of unbiased molecular dynamics. The knotting folding pathway was a large and ambitious conformational change by the standards of protein folding simulations[4]. Analysis of the resulting trajectories indicate that no significant transitions occurred, and in particular, there were no transitions between the separate seeds of the trajectories, suggesting that each 'tree' of sampling was trapped exploring local minima. The high-level schematic illustrations of protein folding landscapes as funnels (such as that depicted in Figure 7.1) obfuscate the fact that many local minima must be traversed on the path to protein folding. The knotting path of MJ0366, in particular, appears to have a rate-limiting step of

the threading event[240].

The trajectories produced in the sampling the loop motion of cyclophilin A, on the other hand, did exhibit transitions between some of the initial conditions. This is because the loop motions are smaller reconfigurations, likely with lower energy barriers, compared to a knotting event. The distance between these configurations in phase space, and thus the likelihood of transition events being sampled, was much greater. However, the cyclophilin A trajectories did not produce any additional samples of several of the initial configurations sampled with interactive molecular dynamics. Instead, the initial conditions quickly relaxed into different minima, from which it struggled to escape. This is due to protein landscapes having many metastable states with a range of barriers between them that are accessed at different timescales[253]. In both systems, due to the lack of converged sampling or a good Markov model, it is impossible to determine whether the pathways identified with interactive molecular dynamics were reasonable, or indeed whether the states that *were* sampled are relevant.

This is the pitfall of unbiased molecular sampling and further evidence of the rare event problem. Without statistically converged observations, quantitative results and conclusions about the dynamics cannot be drawn. The Markov state model framework provides an excellent scheme for quantitatively assessing molecular dynamics trajectories[66]. However, there are many hyper-parameters, which, especially when combined with poorly converged sampling, can be difficult to navigate systematically. In the studies presented here, initial Markov models were built from a limited exploration of the hyper-parameter space. While it is likely that hyper-parameter optimisation, with cross validation, would lead to better Markov models, the ones constructed here served their purpose for providing qualitative insight into the sampling process.

Of course, one could *always* perform more sampling to converge a Markov model. The 10s of microseconds of molecular dynamics undertaken in these applications is a relatively small amount of molecular dynamics by the standards of large compute facilities, which often perform simulations for total aggregate times of hundreds of microseconds or even milliseconds[51, 104, 123]. The problem is that to access rare events, one must perform a sufficient amount of sampling in order to overcome the energetic barriers between disparate states and traverse these barriers a sufficient number of times to converge statistics. While large compute facilities may be able to perform enough molecular dynamics to ignore these inefficiencies, it seems wasteful to run simulations which primarily sample energy minima already observed.

The studies undertaken here sought to investigate whether adaptive sampling, an

attractive strategy that promises automation and statistical rigour, would result in significant acceleration. The results suggest this is not the case, especially compared to the acceleration achieved with the biasing methods of BXD and metadynamics discussed in other chapters. While the adaptive sampling methods based on unbiased molecular dynamics are efficient at sampling systems characterised by small energy barriers, there is no mechanism to guide the system over large energy barriers efficiently. The cluster-based approach developed here enables sampling from multiple disparate initial conditions, and makes sampling completely automated, but cannot directly encourage transitions. A possible improvement would be to define some of the configurations from iMD-VR as targets for other configurations, biasing the sampling to head towards them. This more targeted exploration scheme has been proposed previously by Zimmerman and Bowman in Ref [250], in which the macrostate approach of HTMD can be combined with a specific goal (such as a target RMSD for example).

As noted above, a goal for the follow-up sampling is to provide a means for unbiasing the trajectories produced in interactive molecular dynamics. This is both the bias applied directly by the biasing potentials and the user bias in their hypothesis of a potentially viable pathway. The results of these two studies suggest that performing unbiased molecular dynamics is not an efficient way to do this. Instead, it becomes clear that a more targeted strategy is required that can quickly assess whether a user's intuition was reasonable, and if not, identify alternatives. The accelerated sampling methods such as BXD and the path metadynamics method explored in Chapters 3 and 6 seem to be more suitable for this task.

In both the MJ0366 and cyclophilin A systems, the pathways found using iMD-VR should serve to provide BXD with the intial pathways it needs to accelerate dynamics. At the time of this study an implementation of BXD with OpenMM was not available and so BXD could not be applied to the system. At the time of writing, however, an implementation of BXD with OpenMM has now been developed[4], and it ought to be possible to perform a follow-up study evaluating the iMD-VR paths with BXD.

By providing a set of configurations or an initial path from which appropriate collective variables can be extracted, the interactive molecular dynamics framework enables these methods to be used more easily. The accelerated sampling methods can quickly determine whether these paths are reasonable. In the case of a good pathway, they will be able to produce converged sampling and free energies along the pathway from which quantitative statements about the path may be drawn. In the case of a poor pathway,

[4]Available at `https://github.com/RobinShannon/ChemDyME`.

they may still succeed to sample the path, and thus provide feedback by indicating that the pathway was high in energy. Even in the case that the accelerated sampling fails, information can still be gained. The exhibition of hysteresis - getting trapped in a region of configuration space - is a common way for these methods to fail, and is an indication of a poor path. In a biased simulation based on collective variables, it is usually obvious when this is occurring, as progress along some collective variables is stifled. Compared to unbiased sampling, in which any of these possibilities could also occur, such methods provide rapid evaluation of hypothetical pathways.

CHAPTER 8

CONCLUSIONS AND OUTLOOK

The problem of rare event sampling in molecular dynamics limits the ability of computational scientists to produce converged, statistically significant sampling of molecular processes, particularly long processes such as chemical reactions, enzyme catalysis and protein folding. The contributions of this thesis are to offer some new and updated tools in alleviating this problem.

The boxed molecular dynamics method was automated and generalised to multi-dimensional collective variable space, through the introduction of a general velocity inversion procedure using non-holonomic constraints. These extensions make it applicable to many more systems, enabling the BXD method to take its place in the toolbox of methods for rare event acceleration. It is particularly useful in cases where the dynamics of the system need to be minimally perturbed, in non-equilibrium regimes or where kinetics are the observable that is desired. The contributions of this thesis have since led to the method being extended further for use as a new reaction discovery method in the field of combustion chemistry[124]. However, like other collective variable based methods, its main limitation is the need to identify a suitable set of collective variables which produces a dynamical pathway of the given process that can be accelerated.

The requirement for preliminary exploration of a molecular system in order to identify potential configurations, pathways and collective variables, along with the advent of commodity virtual reality hardware, motivated the development of a framework for interactive molecular dynamics (iMD-VR). The iMD-VR framework developed is a spiritual successor to previous attempts such as VMD IMD that predated commodity VR. By

leveraging modern GPU-accelerated molecular dynamics programs such as OpenMM, cloud-based computing and user-friendly tools such as Unity3D for developing VR applications, the resulting framework is modular, extensible, inherently multi-user and capable of performing simulations in real-time at scales appropriate for applications across multiple fields such as drug discovery and materials. The software is open source and maintained by a growing community of developers.

In the development of the iMD-VR framework novel user interfaces and algorithms were developed to enable rapid and precise manipulation of molecular structures, such as a virtual reality user interface for selecting molecular structures, group interactive potentials and a velocity re-initialisation scheme for fine control.

While the framework is in a state ready for use, there are many improvements to be made. The performance of the system needs improving to minimise overheads in running the simulation, and more advanced rendering techniques need to be used to improve rendering capabilities. Additionally, methods for connecting to existing molecular simulation software packages need to be simplified and standardised, to make it easy for the community to develop extensions. The interaction controls developed so far have been focussed on the manipulation of all-atom biomolecular simulations. One can expect novel interaction methods to be developed as new applications emerge.

Additionally, virtual, augmented and mixed reality hardware solutions continue to be developed. The framework developed is agnostic to the details of these platforms, assuming only that a molecular simulation can be represented in 3D space, and that some form of 3D interaction method allows one to select and interact with the molecules. It should thus be easy to port to new developments as they come to market.

It was demonstrated that by utilising the 3D manipulation enabled by VR, users could reproducibly perform complex molecular manipulations in real-time. The framework thus appears to be appropriate as a method for finding novel configurations and pathways.

Furthermore, because a rigorous molecular simulation is governing the dynamics, continuously driving the system towards stable configurations, the pathways found typically represent reasonable starting points for qualitative insight and follow-up analysis.

This was demonstrated by generating an isomerisation path in the benchmark system alanine dipeptide, in which the pathway found in VR and optimised with the nudged elastic band method was lower in energy than a pathway optimised starting from a linear interpolated structure. By performing metadynamics using the pathway

generated in virtual reality as a collective variable, it was demonstrated that the interactive molecular dynamics framework could be easily combined with existing methods to go from an initial hypothesis through to converged observables, in this case, a free energy surface. While this demonstration did not quite achieve the goal of a completely automated workflow, as the methods used require some hand-tuning, it can be concluded that accelerated sampling from interactive molecular dynamics can produce quantifiable results.

Following the success on a benchmark system, larger systems with more complex transitions were explored. These consisted of knot formation in the protein MJ0366, and loop motions in cyclophilin A. In these applications, it was possible to achieve all the desired transitions and configurations with iMD-VR, but converging follow-up sampling, with the amount of computation time available, was unsuccessful. In particular, it was not possible to quantitatively assess the relevance of pathways and configurations produced with interactive molecular dynamics with the calculation of free energies or rates. The user study of small molecular tasks, the alanine dipeptide example, and the reversibility of the loop motions performed in the cyclophilin A system demonstrate that for simple tasks the pathways produced with interactive molecular dynamics can be reasonable. However, the limits of the kind of processes that can be studied with iMD-VR remains an open question. An application that could test this more thoroughly would be to interactively fold small proteins, such as the β-hairpin[118]. These small proteins are extremely well studied, and so the pathways found through interactive molecular dynamics can be compared to those found with other methods.

The success of the path metadynamics approach for verifying and converging a free energy surface from an iMD-VR simulation, in contrast to the uncertainty from running adaptive sampling with Markov state models, leads to the conclusion that interactive molecular dynamics simulations should be combined with efficient biased sampling so that hypotheses can be rapidly evaluated. Thus, tools for performing dimensionality reduction, extracting paths and sets of collective variables that can be used with accelerated sampling methods such as metadynamics or BXD should be the focus of future development. The path collective variable approach in particular is promising because it includes a mechanism for allowing exploration of the system away from the predetermined path, which makes it possible to sample additional pathways not identified with iMD-VR. This is important as there is no guarantee that a pathway found with iMD-VR will be optimal.

Since its release as an open-source project, Narupa has begun to see uptake both

as a tool for rare event acceleration and for general use as a platform for developing virtual reality enabled interactive simulations. In some recent work in the field of drug discovery, ligand binding simulations have shown binding poses discovered through interactive molecular dynamics simulations are stable in follow-up simulations, providing more evidence that interactive molecular dynamics can produce quantifiable results. It is also being used to generate initial training sets for the machine-learning of potential energy and forces in non-equilibrium reactive systems. In future research, the platform will also be used to facilitate and visualise protein design in collaboration with synthetic biologists.

With the commercial availability of virtual reality hardware for consumers, and the compatibility of the iMD-VR framework with cloud computing, the prospect of 'gamifying' rare event problems becomes tenable. This idea has already been demonstrated to be viable in the field of protein folding[167]. The models available in the iMD-VR framework expand the possibility space for such gamification. Since the framework is based on molecular dynamics and is agnostic to the underlying details of the force fields used, it can be used in a wide variety of fields.

Using the interactive molecular dynamics framework for this purpose is now being explored. In a preliminary application, users can explore chemical reactions in simulations running semi-empirical force fields developed by Reiher and co-workers[162]. Using the interactive force fields, they can bring reactants together, and stretch, break and form bonds to achieve new reactions. Methods for providing feedback and scores of a user's progress in discovering novel reactions are in development.

The maturity of some molecular simulation software, particularly biomolecular simulation, makes it possible to envisage the creation of robust, user-facing applications. For example, in future work, the real-time all-atom models of drug binding simulations will be used to gamify drug discovery. The proposed workflow is to publicly host simulations HPC architecture in the cloud, and drawing inspiration from FoldIt[167], provide 'puzzles' consisting of a set of possible drugs along with a target binding site in a protein. Scoring functions, such as RMSD from the drug candidate to the active site, the potential energy, and other heuristics will be used to provide scoring, as well as visual and auditory feedback. Well-studied examples such as benzylpenicillin binding to beta-lactamase will be used to provide tutorials, but in advanced cases, users will be able to select novel drug candidates.

The updated boxed molecular dynamics algorithm and the interactive molecular dynamics framework are not panaceas to sampling problems in molecular simulation, but

they can alleviate some of the issues, and provide new ways to approach molecular simulation. The variety of applications and ongoing projects that are now beginning to result from this work provide evidence that there is an appetite for these new methods.

APPENDIX A

Example Derivation of the Velocity Inversion Procedure in Boxed Molecular Dynamics

What follows is an example calculation of the velocity inversion procedure using constraints described in Chapter 3, for a simple but illustrative case (the same as that used in Figure 3.6). This example is based on that described by the author in the SI of Ref [131].

Consider a system of atoms A, B, and C where the collective variables are the distances AB and BC. This style of collective variable is useful in many situations, including the acceleration of abstraction reactions as discussed in the main document.

In the interest of saving paper, the example is restricted to two spatial coordinates. Let $\vec{r} = \left[a_x, a_y, b_x, b_y, c_x, c_y\right]$ be the coordinates of the atoms in the system, let $\vec{v} = \left[v_x^a, v_y^a, v_x^b, v_y^b, v_x^c, v_y^c\right]$ be the velocities of atoms A, B and C, and let $\mathbf{M}$ be the diagonal matrix of atomic masses, i.e.:

$$\mathbf{M} = \begin{bmatrix} m_a & 0 & 0 & 0 & 0 & 0 \\ 0 & m_a & 0 & 0 & 0 & 0 \\ 0 & 0 & m_b & 0 & 0 & 0 \\ 0 & 0 & 0 & m_b & 0 & 0 \\ 0 & 0 & 0 & 0 & m_c & 0 \\ 0 & 0 & 0 & 0 & 0 & m_c \end{bmatrix}.$$

The collective variable $\vec{s}(\vec{r})$ representing the distances AB and AC is given by

$$\vec{s}(\vec{r}) = (r_{AB}, r_{BC}), \text{ where} \tag{A.1}$$

$$r_{AB} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}, \tag{A.2}$$

$$r_{BC} = \sqrt{(b_x - c_x)^2 + (b_y - c_y)^2}. \tag{A.3}$$

Suppose there is some BXD boundary B, defined as a two-dimensional line in Hessian form with norm $\vec{n} = (n_1, n_2)$ and point D. The BXD constraint on the dynamics is then

$$\phi = n_1 r_{AB} + n_2 r_{BC} + D \geq 0.$$

Suppose that at some time step the constraint will no longer be satisfied: taking a step forward using the current velocities will result in the boundary being crossed. This is the case in which the velocity reflection is required. In order to perform the velocity reflection using a Lagrangian multiplier, it is necessary to compute $\nabla\phi$, which is given by

$$\nabla\phi^T = \frac{\mathrm{d}\phi}{\mathrm{d}\vec{r}} = n_1 \frac{\mathrm{d}r_{AB}}{\mathrm{d}\vec{r}} + n_2 \frac{\mathrm{d}r_{BC}}{\mathrm{d}\vec{r}} = \begin{bmatrix} n_1(a_x - b_x)/r_{AB} \\ n_1(a_y - b_y)/r_{AB} \\ n_1(a_x - b_x)/r_{AB} + n_2(b_x - c_x)/r_{BC} \\ n_1(a_y - b_y)/r_{AB} + n_2(b_y - c_y)/r_{BC} \\ n_2(c_x - b_x)/r_{BC} \\ n_2(c_y - b_y)/r_{BC} \end{bmatrix}.$$

The expression above demonstrates how the reflection procedure can be constructed from the gradients of the individual components of the collective variables. With $\nabla\phi$ in hand, the Lagrangian multiplier λ can be computed as in Equation 3.16:

$$\lambda = \frac{-2\nabla\phi \cdot \vec{v}}{\nabla\phi \mathbf{M}^{-1} \nabla\phi^T}, \tag{A.4}$$

and subsequently used to compute inverted velocities as $\vec{v}' = \vec{v} + \lambda \mathbf{M}^{-1} \nabla\phi^T$. In the resulting velocities, the components normal to the boundary are inverted, and thus the constraint is satisfied.

APPENDIX B

HIGH PERFORMANCE IMPLEMENTATIONS OF THE MULTI-STATE EMPIRICAL VALENCE BOND METHOD

In this section, more details of the EVB method for the simulation of reactive events are given, and the implementation details of the method used in the simulation of reactions in liquids in Chapter 3.4 are discussed. This section is adapted from the chapter written by Harvey, Glowacki and O'Connor in Ref [45]. In this work, I contributed to the development of the MPI implementation of EVB in CHARMM, performed benchmarks, and wrote a GPU accelerated implementation of EVB developed for use in interactive molecular dynamics (in a precursor application to the NarupaXR framework described in Chapter 4.2)[148].

B.1 Empirical Valence Bond Methods for Exploring Reaction Dynamics in Gas Phase and In Solution

In the EVB reactive dynamics method, a pseudo-Hamiltonian matrix $\mathbf{H}(\vec{r})$, constructed from a reactant function (R) and a product function (P):

$$\mathbf{H}(\vec{r}) = \begin{bmatrix} V_R + e_R & H_{12} \\ H_{12} & V_P + e_P \end{bmatrix},$$

where V_R and V_P are the energies of the reactant and product states respectively, e_R and e_P are constant energy shifts, and H_{12} is a coupling function that couples the reac-

tant and product states. This function is usually implemented as some simple function of nuclear coordinates[145]. This Hamiltonian matrix $\mathbf{H}$ is then diagonalized

$$\mathbf{D} = \mathbf{U}^T \mathbf{H} \mathbf{U} \tag{B.1}$$

to produce $\mathbf{D}$, the diagonal matrix containing the eigenvalues λ_i of $\mathbf{H}$, where $i \in \{0,1\}$. The matrix $\mathbf{U}$ contains the corresponding eigenvectors, $\vec{u}_i$. The ground state energy is taken as λ_0, the lowest eigenvalue of $\mathbf{H}$ (illustrated in Figure B.1), and the coefficients of the corresponding eigenvector $\vec{u}_0$ describe how each state contributes to the state with energy λ_i. The Hellman-Feynman relation is used to produce the set of atomic forces $\mathbf{F}$ for each state:

$$\mathbf{F} = -\frac{\mathrm{d}\mathbf{D}}{\mathrm{d}\vec{r}} = -\mathbf{U}^T \frac{\mathrm{d}\mathbf{H}}{\mathrm{d}\vec{r}} \mathbf{U}. \tag{B.2}$$

The vector $\mathbf{F}_0$ provides the forces corresponding to the lowest eigenvalue λ_0, and are used to propagate the system. To compute $\mathrm{d}\mathbf{H}/\mathrm{d}\vec{r}$, the gradients of V_R and V_P must be computed, along with the gradient of H_{12}. This is a rather convenient calculation as the gradients of V_R and V_P are simply the negative of the forces computed in a typical molecular dynamics calculation. The method described above generalises straightforwardly to the case of many more states, as in the CD_3CN application of Chapter 3.4, in which a flourine radical, which abstracted deuterium from one particular CD_3CN was embedded in n solvent molecules, resulting in a matrix with dimensions $(n+2)\times(n+2)$:

$$\mathbf{H} = \begin{bmatrix} V_1+\epsilon_1 & H_{12} & 0 & 0 & \dots & 0 \\ H_{12} & V_2+\epsilon_2 & H_{23} & H_{24} & \dots & H_{2n} \\ 0 & H_{23} & V_3+\epsilon_3 & 0 & 0 & 0 \\ 0 & H_{24} & 0 & V_4+\epsilon_4 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & H_{2n} & 0 & 0 & \dots & V_{n+2}+\epsilon_{n+2} \end{bmatrix},$$

where V_i is the energy of each state, and there is a coupling term between states 1 and 2 (the reactant and product), and between state 2 and each solvent state representing a proton transfer between the product and a solvent molecule. For a solvent of 62 CD_3CN molecules, resulting in 64 states, this results in a considerable amount molecular dynamics to compute for each state. To be able to perform simulations of this size efficiently, high performance implementations are required.

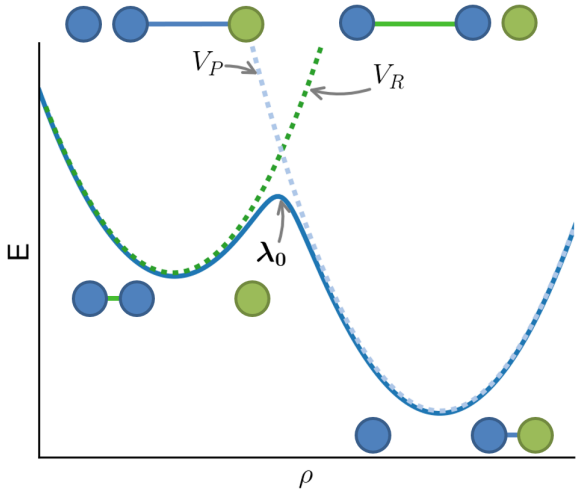


Figure B.1: Illustration of the EVB potential for an abstraction reaction along some reaction coordinate ρ. The potential for the reactant state is shown as the green dashed line, while the potential for the product state is shown as the blue dashed line. The resulting potential formed from the smallest eigenvalue λ_0 is shown in blue, producing a smooth reactive potential.

B.2 High Performance Implementation Details

In the interest of using EVB for large reactive systems, two different implementation strategies have been explored. The first, described here as it is relevant to the applications of Chapter 3.4 , is based on the message-passing interface (MPI) and exploits multi-core computer processing unit (CPU) architectures. A GPU implementation was also developed, the details of which can be found in Ref [45].

The parallelisation strategy used for the CHARMM implementation of EVB is a master/slave approach using MPI, illustrated in Figure B.2. An MPI process manages each state, and propagates the energy and forces applied to that state independently. The master process then gathers together the results for each state to construct the diagonal matrix elements for **H**, then calculates the off-diagonal matrix elements of **H**. With the matrix constructed, it is then diagonalized to solve Equation B.1, yielding the λ_0 eigenvalue of the D matrix and its corresponding eigenvector $\vec{u}_0$. Finally, the master process solves Equation B.2 to yield the Hellman-Feynman force vector $\mathbf{F}_0$. Once this is done, $\mathbf{F}_0$ and λ_0 are then dispatched to each MPI process. Each process then propagates forward a single dynamical timestep, with the identical forces and energies ensuring that the new geometry on each process is also identical. Each process then carries out its own energy and force calculations, the results of which are specific to the connectivity

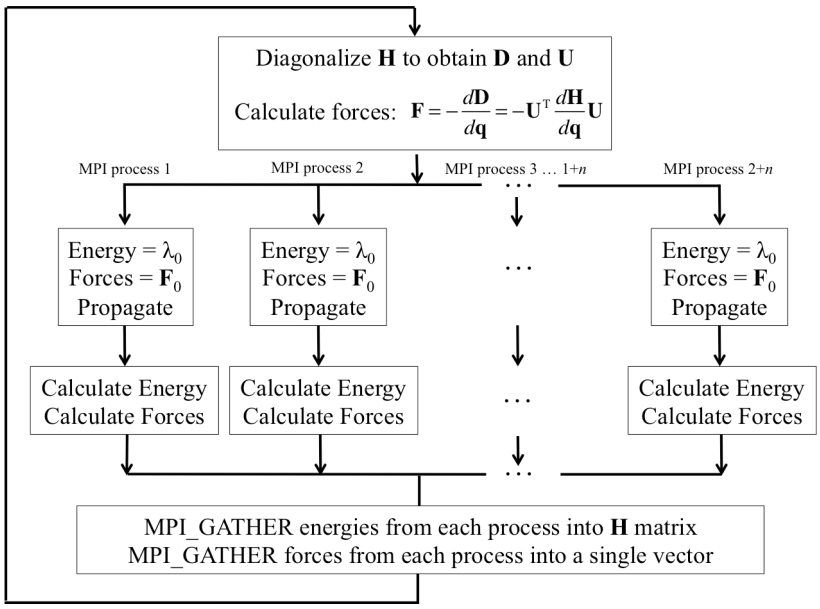


Figure B.2: EVB Propagation scheme using MPI.

of the particular state.

This parallelized propagation strategy scales nearly linearly in the number of EVB states, with the exception of the communication and diagonalisation step during which all processes except the master process are idle. One could improve upon this by introducing parallelism into this step. Figure B.3 shows the scaling that has been observed for the MPI-parallelized CHARMM implementation on up to 64 cores (i.e., eight 8-core nodes), tested on the 64-state F + CD_3CN systems described above, along with analysis of the timings spent on particular computational tasks. For very large matrices, diagonalization will eventually emerge as the computational bottleneck; however, this has not yet been observed for the size of systems the framework has been applied to.

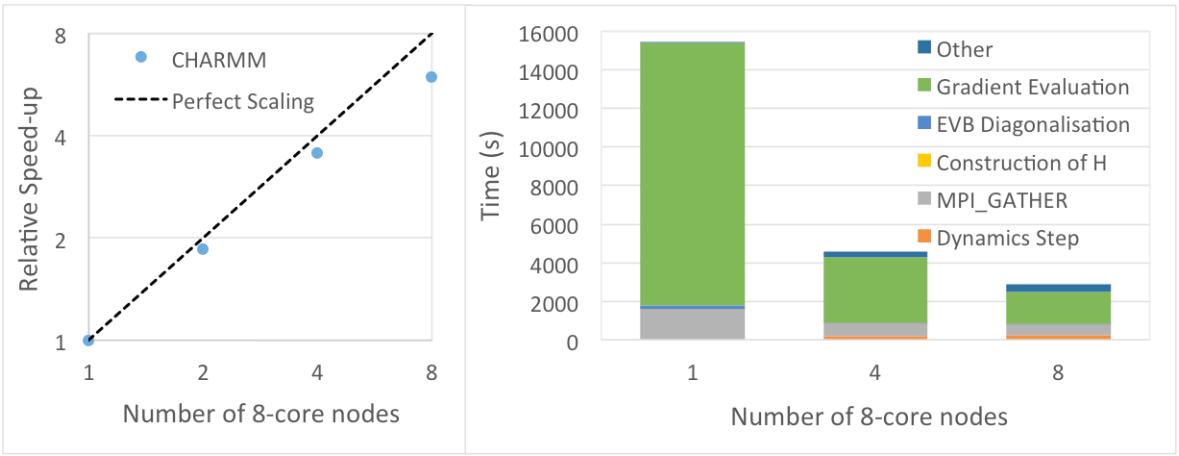


Figure B.3: Performance of the EVB MPI implementation in CHARMM. The left-hand panel shows the relative speed-up in strong scaling, with increasing number of nodes applied to the same 64-state F + CD_3CN system. The right-hand panel shows how the breakdown of time spent on various operations changes as the number of nodes used increases.

APPENDIX C

ERROR ANALYSIS IN BOXED MOLECULAR DYNAMICS

C.1 Block Averaging Analysis for Mean First Passage Times

The calculation of the standard deviation or standard error of a statistic, such as a mean, relies upon the central limit theorem, which holds only for identically distributed independent samples. In a molecular dynamics trajectory, this may not be the case, with samples being correlated between frames. A robust method for accounting for correlation effects is block averaging[254]. For a set of N samples with mean μ, rather than computing the sample variance of all samples of an observed value, the data is instead split into M blocks, the means of each block $\mu_i, i \in [1, M]$ is computed and the variance is computed via

$$\sigma^2 = \frac{1}{M-1} \sum_{i=1}^{M} (\mu_i - \mu)^2,$$

from which error can be calculated as $\sqrt{\sigma^2/M}$. If the block length is 1, meaning $M = N$, then this is simply the usual standard error calculation. However, as the block length increases, the block average accounts for correlation in the data, effectively reducing the number of samples accordingly. This procedure can be used in the calculation of error bars for mean first passage times between boxes in the analysis of a BXD trajectory, accounting for correlation effects between passage times in a robust manner. Figure C.1 shows the error in MFPT with increasing block lengths for two different boxes in the

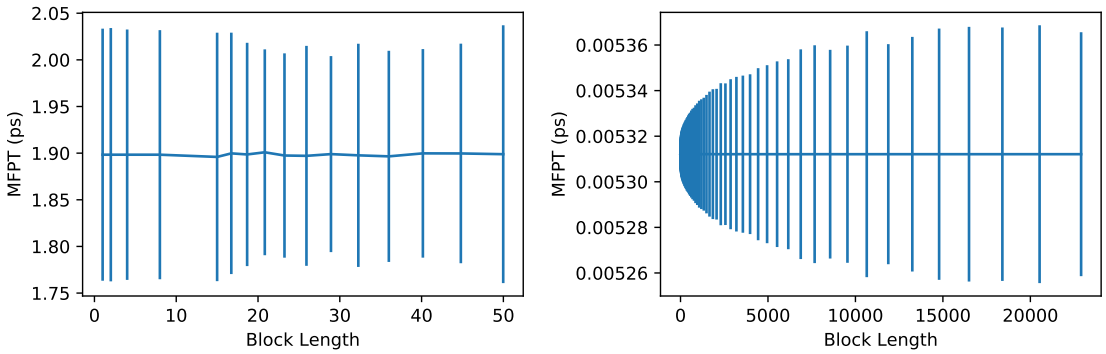


Figure C.1: Block Averaging Analysis for MFPT calculation in a BXD trajectory. The left hand panel shows an MFPT for which passage times were long and therefore decorrelated, as indicated by lack of change in the error bars with increased block length, while the right hand panel shows an box for which passage times were short and correlated.

F + CD_3CN BXD trajectory in solution. The left hand panel corresponds to the passage time from the first box to the second. The error does not increase with block length, indicating that the passage times were not correlated. This is unsurprising given that it is a relatively flat region of the free energy surface with long passage times. The right hand panel, however, corresponding to a box on the steep post-reaction side of the free energy landscape, exhibits highly correlated passage times, as indicated by the error increasing with block length until a converging value is reached at a block length of around 11000. The converged value can then be taken as the error.

C.2 Propagation of Error to Box-to-box Free Energies

The box free energy difference in a BXD calculation is given by

$$\frac{k_{i-1,i}}{k_{i,i-1}} = \exp\left(\frac{-\Delta G_{i-1,i}}{k_B T}\right).$$

A simple rearrangement (dropping k_B and T for convenience) gives

$$\Delta G_{i-1,i} = -\ln\left(\frac{k_{i-1,i}}{k_{i,i-1}}\right).$$

The rates $k_{i,i-1}$ and $k_{i-1,i}$ are given by the calculations of MFPTs $\mu_{i,i-1}$ and $\mu_{i-1,i}$ via $1/\mu_{i,i-1}$ and $1/\mu_{i-1,i}$. The MFPTs are thus the ultimate source of error in a BXD free energy calculation. Let $\sigma_{i,i-1}$ and $\sigma_{i-1,i}$ be the standard error of the MFPTs in the

box. By assuming that $\mu_{i,i-1}$ and $\mu_0 i-1,i$ are independent, we may apply the standard propagation of error formula[255] to get the box-to-box error as

$$\sigma^2_{\Delta G_{i-1,i}} = \left(\frac{\partial \Delta G_{i-1,i}}{\partial \mu_{i-1,i}}\right)^2 \sigma^2_{i-1,i} + \left(\frac{\partial \Delta G_{i-1,i}}{\partial \mu_{i,i-1}}\right)^2 \sigma^2_{i,i-1} \tag{C.1}$$

$$= \left(\frac{\partial}{\partial \mu_{i-1,i}}\left[-\ln\left(\frac{\mu_{i,i-1}}{\mu_{i-1,i}}\right)\right]\right)^2 \sigma^2_{i-1,i} + \left(\frac{\partial}{\partial \mu_{i,i-1}}\left[-\ln\left(\frac{\mu_{i,i-1}}{\mu_{i-1,i}}\right)\right]\right)^2 \sigma^2_{i,i-1} \tag{C.2}$$

$$= \left(\frac{1}{\mu_{i-1,i}}\right)^2 \sigma^2_{i-1,i} + \left(-\frac{1}{\mu_{i,i-1}}\right)^2 \sigma^2_{i,i-1} \tag{C.3}$$

$$= k^2_{i-1,i}\sigma^2_{i-1,i} + k^2_{i,i-1}\sigma^2_{i,i-1} \tag{C.4}$$

This result provides a particular box to box error, but the full box free energy calculation is the accumulation of all prior box free energies:

$$\Delta G_i = \sum_{n=0}^{i} \Delta G_{n-1,n}.$$

This leads to a rather complex looking formula for the propagation of error, as we need to propagate each sampled variable, the MFPTs, for all boxes, but it simplifies nicely:

$$\sigma^2_{\Delta G_i} = \sum_{n=0}^{i}\left[\left(\frac{\partial}{\partial \mu_{n-1,n}}\sum_{j=0}^{i}\Delta G_{j-1,j}\right)^2 \sigma^2_{n-1,n} + \left(\frac{\partial}{\partial \mu_{n,n-1}}\sum_{j=0}^{i}\Delta G_{j-1,j}\right)^2 \sigma^2_{n,n-1}\right] \tag{C.5}$$

$$= \sum_{n=0}^{i} k^2_{n-1,n}\sigma^2_{n-1,n} + k^2_{n,n-1}\sigma^2_{n,n-1} \tag{C.6}$$

$$= \sum_{n=0}^{i} \sigma^2_{\Delta G_{n-1,n}} \tag{C.7}$$

This leads to the expected result that the error in the free energy increases as one proceeds along the boxes, as the error of each previous box is accumulated into the calculation.

BIBLIOGRAPHY

[1] EPCC. *ARCHER » Application usage over past month*. 2018. URL: http://www.archer.ac.uk/status/codes/ (visited on 11/01/2018).

[2] A. Turner and S. Mcintosh-Smith. *A survey of application memory usage on a national supercomputer: an analysis of memory requirements on ARCHER*. Tech. rep. 2017.

[3] A. R. Leach. *Molecular modelling: principles and applications*. 2nd. Pearson Education, 2001, p. 744. ISBN: 0582382106. DOI: qd480.l432001.

[4] S. Piana, K. Lindorff-Larsen, and D. E. Shaw. "Protein folding kinetics and thermodynamics from atomistic simulation". In: *Proceedings of the National Academy of Sciences* 109.44 (Oct. 2012), pp. 17845–17850. ISSN: 0027-8424. DOI: 10.1073/pnas.1201811109.

[5] F. Claeyssens et al. "High-accuracy computation of reaction barriers in enzymes". In: *Angewandte Chemie - International Edition* 45.41 (Oct. 2006), pp. 6856–6859. ISSN: 14337851. DOI: 10.1002/anie.200602711.

[6] G. T. Dunning et al. "Vibrational relaxation and microsolvation of DF after F-atom reactions in polar solvents". In: *Science* 347.6221 (Jan. 2015), pp. 530–533. ISSN: 0036-8075. DOI: 10.1126/science.aaa0103.

[7] H. B. Fan and M. M. F. Yuen. "Material properties of the cross-linked epoxy resin compound predicted by molecular dynamics simulation". In: *Polymer* 48.7 (2007), pp. 2174–2178. ISSN: 0032-3861. DOI: 10.1016/j.polymer.2007.02.007.

[8] S. Doerr et al. "HTMD: High-Throughput Molecular Dynamics for Molecular Discovery". In: *Journal of Chemical Theory and Computation* 12.4 (Apr. 2016), pp. 1845–1852. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.6b00049.

[9] C. Kutzner et al. "Scaling of the GROMACS 4.6 molecular dynamics code on SuperMUC". In: *Advances in Parallel Computing*. Vol. 25. 2014, pp. 722–727. ISBN: 9781614993803. DOI: 10.3233/978-1-61499-381-0-722.

[10] N. Tchipev et al. “TweTriS: Twenty trillion-atom simulation”. In: *The International Journal of High Performance Computing Applications* (Jan. 2019). ISSN: 1094-3420. DOI: 10.1177/1094342018819741.

[11] D. R. Glowacki et al. “A GPU-accelerated immersive audiovisual framework for interaction with molecular dynamics using consumer depth sensors”. In: *Faraday Discussions* 169 (2014), pp. 63–87. ISSN: 1359-6640. DOI: 10.1039/c4fd00008k.

[12] N. Metropolis et al. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092. ISSN: 0021-9606. DOI: 10.1063/1.1699114.

[13] A. Rahman. “Correlations in the Motion of Atoms in Liquid Argon”. In: *Physical Review* 136.2A (Oct. 1964), A405–A411. ISSN: 0031-899X. DOI: 10.1103/PhysRev.136.A405.

[14] J. Hirschfelder, H. Eyring, and B. Topley. “Reactions Involving Hydrogen Molecules and Atoms”. In: *The Journal of Chemical Physics* 4.3 (Mar. 1936), pp. 170–177. ISSN: 0021-9606. DOI: 10.1063/1.1749815.

[15] B. J. Alder, S. P. Frankel, and V. A. Lewinson. “Radial Distribution Function Calculated by the Monte-Carlo Method for a Hard Sphere Fluid”. In: *The Journal of Chemical Physics* 23.3 (Mar. 1955), pp. 417–419. ISSN: 0021-9606. DOI: 10.1063/1.1742004.

[16] W. W. Wood and J. D. Jacobson. “Preliminary Results from a Recalculation of the Monte Carlo Equation of State of Hard Spheres”. In: *The Journal of Chemical Physics* 27.5 (Nov. 1957), pp. 1207–1208. ISSN: 0021-9606. DOI: 10.1063/1.1743956.

[17] A. Rahman and F. H. Stillinger. “Molecular Dynamics Study of Liquid Water”. In: *The Journal of Chemical Physics* 55.7 (Oct. 1971), pp. 3336–3359. ISSN: 0021-9606. DOI: 10.1063/1.1676585.

[18] M. Levitt and S. Lifson. “Refinement of protein conformations using a macromolecular energy minimization procedure”. In: *Journal of Molecular Biology* 46.2 (Dec. 1969), pp. 269–279. ISSN: 00222836. DOI: 10.1016/0022-2836(69)90421-5.

[19] A. Warshel and M. Levitt. “Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme”. In: *Journal of Molecular Biology* 103.2 (May 1976), pp. 227–249. ISSN: 00222836. DOI: 10.1016/0022-2836(76)90311-9.

[20] B. R. Gelin and M. Karplus. “Sidechain torsional potentials and motion of amino acids in porteins: bovine pancreatic trypsin inhibitor.” In: *Proceedings of the National Academy of Sciences* 72.6 (June 1975), pp. 2002–2006. ISSN: 0027-8424. DOI: 10.1073/pnas.72.6.2002.

[21] B. R. Gelin. “Application of empirical energy functions to conformational problems in biochemical systems”. PhD thesis. Harvard University, 1976.

[22] J. A. McCammon, B. R. Gelin, and M. Karplus. “Dynamics of folded proteins.” In: *Nature* 267.5612 (June 1977), pp. 585–90. ISSN: 0028-0836.

[23] F. Vitalini et al. “Dynamic properties of force fields”. In: *Journal of Chemical Physics* 142.8 (Feb. 2015), pp. 1–12. ISSN: 0021-9606. DOI: 10.1063/1.4909549.

[24] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications*. Vol. 1. Academic press, 2001.

[25] R. J. Bartlett and M. Musiał. “Coupled-cluster theory in quantum chemistry”. In: *Reviews of Modern Physics* 79.1 (Feb. 2007), pp. 291–352. ISSN: 0034-6861. DOI: 10.1103/RevModPhys.79.291.

[26] K. K. Baeck and T. J. Martinez. “Ab initio molecular dynamics with equation-of-motion coupled-cluster theory: electronic absorption spectrum of ethylene”. In: *Chemical Physics Letters* 375.3-4 (July 2003), pp. 299–308. ISSN: 00092614. DOI: 10.1016/S0009-2614(03)00847-9.

[27] A. R. Leach. “Empirical Force Field Models: Molecular Mechanics”. In: *Molecular modelling: principles and applications*. 201, pp. 165–247.

[28] I. S. Ufimtsev and T. J. Martinez. “Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients, Geometry Optimization, and First Principles Molecular Dynamics”. In: *Journal of Chemical Theory and Computation* 5.10 (Oct. 2009), pp. 2619–2628. ISSN: 1549-9618. DOI: 10.1021/ct9003004.

[29] B. Aradi, B. Hourahine, and T. Frauenheim. *DFTB+, a sparse matrix-based implementation of the DFTB method*. 2007.

[30] P. Koskinen and V. Mäkinen. “Density-functional tight-binding for beginners”. In: *Computational Materials Science* 47.1 (Nov. 2009), pp. 237–253. ISSN: 09270256. DOI: 10.1016/j.commatsci.2009.07.013. arXiv: 0910.5861.

[31] J. J. P. Stewart. “Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements”. In: *Journal of Molecular Modeling* 13.12 (2007), pp. 1173–1213. ISSN: 0948-5023. DOI: 10.1007/s00894-007-0233-4.

[32] S. J. Marrink et al. “The MARTINI force field: coarse grained model for biomolecular simulations”. In: *The journal of physical chemistry B* 111.27 (2007), pp. 7812–7824. ISSN: 1520-6106.

[33] H. M. Senn and W. Thiel. “QM/MM methods for biomolecular systems.” In: *Angewandte Chemie (International ed. in English)* 48.7 (Jan. 2009), pp. 1198–229. ISSN: 1521-3773. DOI: 10.1002/anie.200802019.

[34] H. Lin and D. G. Truhlar. “QM/MM: what have we learned, where are we, and where do we go from here?” In: *Theoretical Chemistry Accounts* 117.2 (July 2006), pp. 185–199. ISSN: 1432-881X. DOI: 10.1007/s00214-006-0143-z.

[35] A. W. Götz, M. A. Clark, and R. C. Walker. “An extensible interface for QM/MM molecular dynamics simulations with AMBER”. In: *Journal of Computational Chemistry* 35.2 (Jan. 2014), pp. 95–108. ISSN: 01928651. DOI: 10.1002/jcc.23444.

[36] J. E. Lennard-Jones. “Cohesion”. In: *Proceedings of the Physical Society* 43.5 (Sept. 1931), pp. 461–482. ISSN: 0959-5309. DOI: 10.1088/0959-5309/43/5/301.

[37] L. Verlet. “Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules”. In: *Physical Review* 159.1 (July 1967), pp. 98–103. ISSN: 0031-899X. DOI: 10.1103/PhysRev.159.98.

[38] T. A. Halgren and W. Damm. “Polarizable force fields”. In: *Current Opinion in Structural Biology* 11.2 (2001), pp. 236–242. ISSN: 0959-440X. DOI: https://doi.org/10.1016/S0959-440X(00)00196-2.

[39] C. M. Baker. “Polarizable force fields for molecular dynamics simulations of biomolecules”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 5.2 (2015), pp. 241–254. ISSN: 17590884. DOI: 10.1002/wcms.1215.

[40] P. Eastman and V. Pande. “OpenMM: A Hardware-Independent Framework for Molecular Simulations”. In: *Computing in Science & Engineering* 12.4 (July 2010), pp. 34–39. ISSN: 1521-9615. DOI: 10.1109/MCSE.2010.27.

[41] R. Hockney and J. Eastwood. *Computer simulation using particles*. CRC Press, 1988.

[42] A. C. T. van Duin et al. “ReaxFF: A Reactive Force Field for Hydrocarbons”. In: *The Journal of Physical Chemistry A* 105.41 (Oct. 2001), pp. 9396–9409. ISSN: 1089-5639. DOI: 10.1021/jp004368u.

[43] J. Aaqvist and A. Warshel. “Simulation of enzyme reactions using valence bond force fields and other hybrid quantum/classical approaches”. In: *Chemical Reviews* 93.7 (Nov. 1993), pp. 2523–2544. ISSN: 0009-2665. DOI: 10.1021/cr00023a010.

[44] A. Warshel and R. M. Weiss. “An empirical valence bond approach for comparing reactions in solutions and in enzymes”. In: *Journal of the American Chemical Society* 102.20 (Sept. 1980), pp. 6218–6226. ISSN: 0002-7863. DOI: 10.1021/ja00540a008.

[45] J. N. Harvey, M. O’Connor, and D. R. Glowacki. “Empirical Valence Bond Methods for Exploring Reaction Dynamics in the Gas Phase and in Solution”. In: *Theory and Applications of the Empirical Valence Bond Approach*. Chichester, UK: John Wiley & Sons, Ltd, Feb. 2017, pp. 93–119. DOI: 10.1002/9781119245544.ch4.

[46] F. Pietrucci. “Strategies for the exploration of free energy landscapes: Unity in diversity and challenges ahead”. In: *Reviews in Physics* 2 (Nov. 2017), pp. 32–45. ISSN: 24054283. DOI: 10.1016/j.revip.2017.05.001.

[47] C. Abrams and G. Bussi. “Enhanced Sampling in Molecular Dynamics Using Metadynamics, Replica-Exchange, and Temperature-Acceleration”. en. In: *Entropy* 16.1 (Dec. 2013), pp. 163–199. ISSN: 1099-4300. DOI: 10.3390/e16010163. arXiv: arXiv:1401.0387v1.

[48] R. Cuchillo, K. Pinto-Gil, and J. Michel. “A Collective Variable for the Rapid Exploration of Protein Druggability”. In: *Journal of Chemical Theory and Computation* 11.3 (Mar. 2015), pp. 1292–1307. ISSN: 1549-9618. DOI: 10.1021/ct501072t.

[49] R. T. McGibbon et al. “MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories”. In: *Biophysical Journal* 109.8 (Oct. 2015), pp. 1528–1532. ISSN: 15420086. DOI: 10.1016/j.bpj.2015.08.015.

[50] M. Shirts and V. S. Pande. “Screen Savers of the World Unite!” In: *Science* 290.5498 (Dec. 2000), pp. 1903–1904. ISSN: 00368075. DOI: 10.1126/science.290.5498.1903.

[51] A. L. Beberg et al. “Folding@home: Lessons from eight years of volunteer distributed computing”. In: *IPDPS 2009 - Proceedings of the 2009 IEEE International Parallel and Distributed Processing Symposium*. 2009. ISBN: 9781424437504. DOI: 10.1109/IPDPS.2009.5160922.

[52] S. M. Larson et al. “Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology”. In: *arxiv.org* (2009), p. 31. arXiv: 0901.0866.

[53] D. E. Shaw et al. “Anton, a special-purpose machine for molecular dynamics simulation”. In: *Communications of the ACM* 51.7 (July 2008), p. 91. ISSN: 00010782. DOI: 10.1145/1364782.1364802.

[54] V. S. Pande, K. Beauchamp, and G. R. Bowman. “Everything you wanted to know about Markov State Models but were afraid to ask”. In: *Methods* 52.1 (Sept. 2010), pp. 99–105. ISSN: 10462023. DOI: 10.1016/j.ymeth.2010.06.002.

[55] J. D. Chodera and F. Noé. “Markov state models of biomolecular conformational dynamics”. In: *Current Opinion in Structural Biology* 25 (Apr. 2014), pp. 135–144. ISSN: 0959440X. DOI: 10.1016/j.sbi.2014.04.002. arXiv: NIHMS150003.

[56] G. R. Bowman et al. “Progress and challenges in the automated construction of Markov state models for full protein systems”. In: *The Journal of Chemical Physics* 131.12 (Sept. 2009), p. 124101. ISSN: 0021-9606. DOI: 10.1063/1.3216567.

[57] G. R. Bowman. “An Overview and Practical Guide to Building Markov State Models”. In: *Advances in Experimental Medicine and Biology*. Vol. 797. Springer, Dordrecht, 2014, pp. 7–22. ISBN: 9789400776050. DOI: 10.1007/978-94-007-7606-7_2.

[58] G. Pérez-Hernández et al. “Identification of slow molecular order parameters for Markov model construction”. In: *The Journal of Chemical Physics* 139.1 (July 2013), p. 015102. ISSN: 0021-9606. DOI: 10.1063/1.4811489.

[59] C. R. Schwantes and V. S. Pande. “Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9”. In: *Journal of Chemical Theory and Computation* 9.4 (Apr. 2013), pp. 2000–2009. ISSN: 1549-9618. DOI: 10.1021/ct300878a.

[60] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2012), pp. 2825–2830. ISSN: 15324435. DOI: 10.1007/s13398-014-0173-7.2. arXiv: 1201.0490.

[61] C. Wehmeyer and F. Noé. “Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics”. In: *The Journal of Chemical Physics* 148.24 (June 2018), p. 241703. ISSN: 0021-9606. DOI: 10.1063/1.5011399. arXiv: 1710.11239.

[62] F. Noé et al. “Projected and hidden Markov models for calculating kinetics and metastable states of complex molecules”. In: *Journal of Chemical Physics* 139.18 (2013), 11B609_1. ISSN: 00219606. DOI: 10.1063/1.4828816. arXiv: arXiv:1309.3220v1.

[63] N. Singhal and V. S. Pande. “Error analysis and efficient sampling in Markovian state models for molecular dynamics”. In: *The Journal of Chemical Physics* 123.20 (Nov. 2005), p. 204909. ISSN: 0021-9606. DOI: 10.1063/1.2116947.

[64] P. Deuflhard and M. Weber. “Robust Perron cluster analysis in conformation dynamics”. In: *Linear Algebra and its Applications* 398.1-3 (Mar. 2005), pp. 161–184. ISSN: 00243795. DOI: 10.1016/j.laa.2004.10.026.

[65] F. Noé and S. Fischer. “Transition networks for modeling the kinetics of conformational change in macromolecules”. In: *Current Opinion in Structural Biology* 18.2 (Apr. 2008), pp. 154–162. ISSN: 0959440X. DOI: 10.1016/j.sbi.2008.01.008.

[66] B. E. Husic and V. S. Pande. “Markov State Models: From an Art to a Science”. In: *Journal of the American Chemical Society* 140.7 (Feb. 2018), pp. 2386–2396. ISSN: 0002-7863. DOI: 10.1021/jacs.7b12191.

[67] M. P. Harrigan et al. “MSMBuilder: Statistical Models for Biomolecular Dynamics”. In: *Biophysical Journal* 112.1 (Jan. 2017), pp. 10–15. ISSN: 00063495. DOI: 10.1016/j.bpj.2016.10.042.

[68] M. K. Scherer et al. “PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models”. In: *Journal of Chemical Theory and Computation* 11.11 (Nov. 2015), pp. 5525–5542. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.5b00743.

[69] D. R. Glowacki, E. Paci, and D. V. Shalashilin. “Boxed Molecular Dynamics: A Simple and General Technique for Accelerating Rare Event Kinetics and Mapping Free Energy in Large Molecular Systems”. In: *The Journal of Physical Chemistry B* 113.52 (Dec. 2009), pp. 16603–16611. ISSN: 1520-6106. DOI: 10.1021/jp9074898.

[70] D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein. “Current Status of Transition-State Theory”. In: *The Journal of Physical Chemistry* 100.31 (Jan. 1996), pp. 12771–12800. ISSN: 0022-3654. DOI: 10.1021/jp953748q.

[71] H. Eyring. “The Activated Complex in Chemical Reactions”. In: *The Journal of Chemical Physics* 3.2 (Feb. 1935), pp. 107–115. ISSN: 0021-9606. DOI: 10.1063/1.1749604.

[72] L. Y. P. Luk et al. “Unraveling the role of protein dynamics in dihydrofolate reductase catalysis”. In: *Proceedings of the National Academy of Sciences* 110.41 (Oct. 2013), pp. 16344–16349. ISSN: 0027-8424. DOI: 10.1073/pnas.1312437110.

[73] D. E. Shaw et al. “Anton 2: Raising the Bar for Performance and Programmability in a Special-Purpose Molecular Dynamics Supercomputer”. In: *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*. January. IEEE, Nov. 2014, pp. 41–53. ISBN: 978-1-4799-5500-8. DOI: 10.1109/SC.2014.9.

[74] A. Stank et al. “Protein Binding Pocket Dynamics”. In: *Accounts of Chemical Research* 49.5 (May 2016), pp. 809–815. ISSN: 0001-4842. DOI: 10.1021/acs.accounts.5b00516.

[75] D. De Sancho, U. Doshi, and V. Muñoz. “Protein Folding Rates and Stability: How Much Is There Beyond Size?” In: *Journal of the American Chemical Society* 131.6 (Feb. 2009), pp. 2074–2075. ISSN: 0002-7863. DOI: 10.1021/ja808843h.

[76] P. E. Leopold, M. Montal, and J. N. Onuchic. “Protein folding funnels: a kinetic approach to the sequence-structure relationship.” In: *Proceedings of the National*

Academy of Sciences of the United States of America 89.18 (Sept. 1992), pp. 8721–5. ISSN: 0027-8424.

[77] A. Geist and D. A. Reed. "A survey of high-performance computing scaling challenges". In: *The International Journal of High Performance Computing Applications* 31.1 (Jan. 2017), pp. 104–113. ISSN: 1094-3420. DOI: 10.1177/1094342015597083.

[78] J. Shalf, S. Dosanjh, and J. Morrison. "Exascale Computing Technology Challenges". In: *High performance computing for computational science – VECPAR 2010. 9th international conference*. 2010, pp. 1–25. DOI: 10.1007/978-3-642-19328-6_1.

[79] L. B. Kish. "End of Moore's law: thermal (noise) death of integration in micro and nano electronics". In: *Physics Letters A* 305.3-4 (Dec. 2002), pp. 144–149. ISSN: 03759601. DOI: 10.1016/S0375-9601(02)01365-8.

[80] A. A. Chien and V. Karamcheti. "Moore's Law: The First Ending and a New Beginning". English. In: *Computer* 46.12 (Dec. 2013), pp. 48–53. ISSN: 0018-9162. DOI: 10.1109/MC.2013.431.

[81] M. Lobet et al. "High-Performance Computing at Exascale: challenges and benefits". In: *15ème congrés de la Société Française de Physique division Plasma*. 2018.

[82] M. S. Friedrichs et al. "Accelerating molecular dynamic simulation on graphics processing units". In: *Journal of Computational Chemistry* 30.6 (Apr. 2009), pp. 864–872. ISSN: 01928651. DOI: 10.1002/jcc.21209.

[83] P. Eastman and V. S. Pande. "Efficient nonbonded interactions for molecular dynamics on a graphics processing unit". In: *Journal of Computational Chemistry* 31.6 (Apr. 2009), NA–NA. ISSN: 01928651. DOI: 10.1002/jcc.21413.

[84] S. Pronk et al. "GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit". In: *Bioinformatics* 29.7 (Apr. 2013), pp. 845–854. ISSN: 1460-2059. DOI: 10.1093/bioinformatics/btt055.

[85] R. Salomon-Ferrer, D. A. Case, and R. C. Walker. "An overview of the Amber biomolecular simulation package". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 3.2 (Mar. 2013), pp. 198–210. ISSN: 17590876. DOI: 10.1002/wcms.1121.

[86] J. C. Phillips et al. “Scalable molecular dynamics with NAMD”. In: *Journal of Computational Chemistry* 26.16 (Dec. 2005), pp. 1781–1802. ISSN: 0192-8651. DOI: 10.1002/jcc.20289.

[87] I. Buch et al. “High-Throughput All-Atom Molecular Dynamics Simulations Using Distributed Computing”. In: *Journal of Chemical Information and Modeling* 50.3 (Mar. 2010), pp. 397–403. ISSN: 1549-9596. DOI: 10.1021/ci900455r.

[88] P. Wapeesittipan et al. “Allosteric effects in a catalytically impaired variant of the enzyme Cyclophilin A are unrelated to millisecond time scale motions”. In: *bioRxiv* (Jan. 2017). DOI: 10.1101/224329.

[89] V. S. Pande and D. S. Rokhsar. “Molecular dynamics simulations of unfolding and refolding of a beta-hairpin fragment of protein G”. In: *Proceedings of the National Academy of Sciences* 96.16 (Aug. 1999), pp. 9062–9067. ISSN: 0027-8424. DOI: 10.1073/pnas.96.16.9062.

[90] Y. Sugita and Y. Okamoto. “Replica-exchange molecular dynamics method for protein folding”. In: *Chemical Physics Letters* 314.1-2 (Nov. 1999), pp. 141–151. ISSN: 00092614. DOI: 10.1016/S0009-2614(99)01123-9.

[91] C. T. Leahy et al. “Peptide dimerization-dissociation rates from replica exchange molecular dynamics”. In: *The Journal of Chemical Physics* 147.15 (Oct. 2017), p. 152725. ISSN: 0021-9606. DOI: 10.1063/1.5004774.

[92] O. Valsson, P. Tiwary, and M. Parrinello. “Enhancing Important Fluctuations: Rare Events and Metadynamics from a Conceptual Viewpoint”. In: *Annual Review of Physical Chemistry* 67.1 (May 2016), pp. 159–184. ISSN: 0066-426X. DOI: 10.1146/annurev-physchem-040215-112229.

[93] P. Liu et al. “Replica exchange with solute tempering: A method for sampling biological systems in explicit water”. In: *Proceedings of the National Academy of Sciences of the United States of America* 102.39 (Sept. 2005), 13749 LP –13754. DOI: 10.1073/pnas.0506346102.

[94] L. Wang, R. A. Friesner, and B. J. Berne. “Replica Exchange with Solute Scaling: A More Efficient Version of Replica Exchange with Solute Tempering (REST2)”. In: *The Journal of Physical Chemistry B* 115.30 (Aug. 2011), pp. 9431–9438. ISSN: 1520-6106. DOI: 10.1021/jp204407d.

[95] P. G. Bolhuis et al. “Transition path sampling: throwing ropes over rough mountain passes, in the dark.” en. In: *Annual review of physical chemistry* 53.1 (Jan. 2002), pp. 291–318. ISSN: 0066-426X. DOI: 10.1146/annurev.physchem.53.082301.113146.

[96] T. S. van Erp and P. G. Bolhuis. “Elaborating transition interface sampling methods”. In: *Journal of Computational Physics* 205.1 (2005), pp. 157–181. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2004.11.003.

[97] R. J. Allen, C. Valeriani, and P. Rein ten Wolde. “Forward flux sampling for rare event simulations”. In: *Journal of Physics: Condensed Matter* 21.46 (Nov. 2009), p. 463102. ISSN: 0953-8984. DOI: 10.1088/0953-8984/21/46/463102.

[98] G. Huber and S. Kim. “Weighted-ensemble Brownian dynamics simulations for protein association reactions”. In: *Biophysical Journal* 70.1 (Jan. 1996), pp. 97–110. ISSN: 00063495. DOI: 10.1016/S0006-3495(96)79552-8.

[99] A. Dickson and C. L. Brooks. “WExplore: Hierarchical Exploration of High-Dimensional Spaces Using the Weighted Ensemble Algorithm”. In: *The Journal of Physical Chemistry B* 118.13 (Apr. 2014), pp. 3532–3542. ISSN: 1520-6106. DOI: 10.1021/jp411479c.

[100] S. D. Lotz and A. Dickson. “Unbiased Molecular Dynamics of 11 min Timescale Drug Unbinding Reveals Transition State Stabilizing Interactions”. In: *Journal of the American Chemical Society* 140.2 (Jan. 2018), pp. 618–628. ISSN: 0002-7863. DOI: 10.1021/jacs.7b08572.

[101] D. R. Glowacki, E. Paci, and D. V. Shalashilin. “Boxed Molecular Dynamics: Decorrelation Time Scales and the Kinetic Master Equation”. In: *Journal of Chemical Theory and Computation* 7.5 (May 2011), pp. 1244–1252. ISSN: 1549-9618. DOI: 10.1021/ct200011e.

[102] A. K. Faradjian and R. Elber. “Computing time scales from reaction coordinates by milestoning”. In: *The Journal of Chemical Physics* 120.23 (June 2004), pp. 10880–10889. ISSN: 0021-9606. DOI: 10.1063/1.1738640.

[103] E. Vanden-Eijnden and M. Venturoli. “Markovian milestoning with Voronoi tessellations”. In: *Journal of Chemical Physics* 130.19 (May 2009), p. 194101. ISSN: 00219606. DOI: 10.1063/1.3129843.

[104] N. Plattner et al. “Complete protein–protein association kinetics in atomic detail revealed by molecular dynamics simulations and Markov modelling”. In: *Nature Chemistry* 9.10 (June 2017), pp. 1005–1011. ISSN: 1755-4330. DOI: 10.1038/nchem.2785.

[105] S. Izrailev et al. “Steered Molecular Dynamics”. In: *Computational Molecular Dynamics: Challenges, Methods, Ideas SE - 2*. Vol. 4. 1999, pp. 39–65. ISBN: 978-3-540-63242-9. DOI: 10.1007/978-3-642-58360-5_2.

[106] J. Schlitter, M. Engels, and P. Krüger. “Targeted molecular dynamics: A new approach for searching pathways of conformational transitions”. In: *Journal of Molecular Graphics* 12.2 (June 1994), pp. 84–89. ISSN: 02637855. DOI: 10.1016/0263-7855(94)80072-3.

[107] J. Schlitter et al. “Targeted Molecular Dynamics Simulation of Conformational Change-Application to the T <-> R Transition in Insulin”. In: *Molecular Simulation* 10.2-6 (Aug. 1993), pp. 291–308. ISSN: 0892-7022. DOI: 10.1080/08927029308022170.

[108] S. Park and K. Schulten. “Calculating potentials of mean force from steered molecular dynamics simulations”. In: *The Journal of Chemical Physics* 120.13 (Apr. 2004), pp. 5946–5961. ISSN: 0021-9606. DOI: 10.1063/1.1651473.

[109] B. Isralewitz, M. Gao, and K. Schulten. “Steered molecular dynamics and mechanical functions of proteins”. In: *Current Opinion in Structural Biology* 11.2 (Apr. 2001), pp. 224–230. ISSN: 0959440X. DOI: 10.1016/S0959-440X(00)00194-9.

[110] G. Torrie and J. Valleau. “Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling”. In: *Journal of Computational Physics* 23.2 (Feb. 1977), pp. 187–199. ISSN: 00219991. DOI: 10.1016/0021-9991(77)90121-8.

[111] M. Souaille and B. Roux. “Extension to the weighted histogram analysis method: combining umbrella sampling with free energy calculations”. In: *Computer Physics Communications* 135.1 (Mar. 2001), pp. 40–57. ISSN: 00104655. DOI: 10.1016/S0010-4655(00)00215-0.

[112] E. Rosta and G. Hummer. “Free Energies from Dynamic Weighted Histogram Analysis Using Unbiased Markov State Model”. English. In: *Journal Of Chemi-*

cal Theory And Computation 11.1 (Jan. 2014). ISSN: 1549-9618. DOI: 10.1021/ct500719p.

[113] M. Mezei. "Adaptive umbrella sampling: Self-consistent determination of the non-Boltzmann bias". In: *Journal of Computational Physics* 68.1 (Jan. 1987), pp. 237–248. ISSN: 00219991. DOI: 10.1016/0021-9991(87)90054-4.

[114] A. Laio and M. Parrinello. "Escaping free-energy minima". In: *Proceedings of the National Academy of Sciences* 99.20 (Oct. 2002), pp. 12562–12566. ISSN: 0027-8424. DOI: 10.1073/pnas.202427399.

[115] PLUMED. *Belfast tutorial: Metadynamics*. URL: https://plumed.github.io/doc-v2.5/user-doc/html/belfast-6.html (visited on 01/08/2019).

[116] A. Barducci, G. Bussi, and M. Parrinello. "Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method". In: *Physical Review Letters* 100.2 (Jan. 2008), p. 020603. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.100.020603.

[117] M. Bonomi et al. "PLUMED: A portable plugin for free-energy calculations with molecular dynamics". In: *Computer Physics Communications* 180.10 (Oct. 2009), pp. 1961–1972. ISSN: 00104655. DOI: 10.1016/j.cpc.2009.05.011.

[118] G. Bussi et al. "Free-Energy Landscape for β Hairpin Folding from Combined Parallel Tempering and Metadynamics". In: *Journal of the American Chemical Society* 128.41 (Oct. 2006), pp. 13435–13441. ISSN: 0002-7863. DOI: 10.1021/ja062463w.

[119] S. Piana and A. Laio. "A Bias-Exchange Approach to Protein Folding". In: *The Journal of Physical Chemistry B* 111.17 (May 2007), pp. 4553–4559. ISSN: 1520-6106. DOI: 10.1021/jp067873l.

[120] M. Bonomi and M. Parrinello. "Enhanced Sampling in the Well-Tempered Ensemble". In: *Physical review letters* 104 (May 2010), p. 190601. DOI: 10.1103/PhysRevLett.104.190601.

[121] D. Hamelberg, J. Mongan, and J. A. McCammon. "Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules". In: *The Journal of chemical physics* 120.24 (2004), pp. 11919–11929. ISSN: 0021-9606.

[122] L. C. T. Pierce et al. "Routine Access to Millisecond Time Scale Events with Accelerated Molecular Dynamics". In: *Journal of chemical theory and computation* 8.9 (Sept. 2012), pp. 2997–3002. ISSN: 1549-9626. DOI: 10.1021/ct300284c.

[123] J. Juarez-Jimenez et al. “Atomic Level Characterisation of Millisecond-Time Scale Protein Motions through a Combined Molecular Simulations and NMR Approach”. In: *bioRxiv* (Dec. 2018), p. 490987. DOI: 10.1101/490987.

[124] R. J. Shannon et al. “Adaptively Accelerating Reactive Molecular Dynamics Using Boxed Molecular Dynamics in Energy Space”. In: *Journal of Chemical Theory and Computation* 14.9 (Sept. 2018), pp. 4541–4552. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.8b00515.

[125] R. R. Coifman et al. “Diffusion Maps, Reduction Coordinates, and Low Dimensional Representation of Stochastic Systems”. In: *Multiscale Modeling & Simulation* 7.2 (Jan. 2008), pp. 842–864. ISSN: 1540-3459. DOI: 10.1137/070696325.

[126] N. Elmaci and R. S. Berry. “Principal coordinate analysis on a protein model”. In: *The Journal of Chemical Physics* 110.21 (June 1999), pp. 10606–10622. ISSN: 0021-9606. DOI: 10.1063/1.478992.

[127] M. M. Sultan and V. S. Pande. “tICA-Metadynamics: Accelerating Metadynamics by Using Kinetically Selected Collective Variables”. In: *Journal of Chemical Theory and Computation* 13.6 (June 2017), pp. 2440–2447. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.7b00182.

[128] R. T. McGibbon, B. E. Husic, and V. S. Pande. “Identification of simple reaction coordinates from complex dynamics”. In: *The Journal of Chemical Physics* 146.4 (Jan. 2017), p. 044109. ISSN: 0021-9606. DOI: 10.1063/1.4974306. arXiv: 1602.08776.

[129] D. Branduardi, F. L. Gervasio, and M. Parrinello. “From A to B in free energy space”. In: *The Journal of Chemical Physics* 126.5 (Feb. 2007), p. 054103. ISSN: 0021-9606. DOI: 10.1063/1.2432340.

[130] S. Brandt et al. “Machine Learning of Biomolecular Reaction Coordinates”. In: *The Journal of Physical Chemistry Letters* 9.9 (May 2018), pp. 2144–2150. ISSN: 1948-7185. DOI: 10.1021/acs.jpclett.8b00759.

[131] M. O’Connor et al. “Adaptive free energy sampling in multidimensional collective variable space using boxed molecular dynamics”. In: *Faraday Discuss.* 195 (Jan. 2016), pp. 395–419. ISSN: 1359-6640. DOI: 10.1039/C6FD00138F.

[132] D. R. Glowacki et al. “Ultrafast energy flow in the wake of solution-phase bimolecular reactions”. In: *Nature Chemistry* 3.11 (Nov. 2011), pp. 850–855. ISSN: 1755-4330. DOI: 10.1038/nchem.1154.

[133] S. J. Greaves et al. “Vibrationally quantum-state-specific reaction dynamics of H atom abstraction by CN radical in solution.” In: *Science (New York, N.Y.)* 331.2011 (Mar. 2011), pp. 1423–6. ISSN: 0036-8075. DOI: 10.1126/science.1197796.

[134] C. Bartels and M. Karplus. “Multidimensional adaptive umbrella sampling: Applications to main chain and side chain peptide conformations”. In: *Journal of Computational Chemistry* 18.12 (Sept. 1997), pp. 1450–1462. ISSN: 0192-8651. DOI: 10.1002/(SICI)1096-987X(199709)18:12<1450::AID-JCC3>3.0.CO;2-I.

[135] P. Lötstedt. “Mechanical Systems of Rigid Bodies Subject to Unilateral Constraints”. In: *SIAM Journal on Applied Mathematics* 42.2 (Apr. 1982), pp. 281–296. ISSN: 0036-1399. DOI: 10.1137/0142022.

[136] P. Lötstedt. “Mechanical Systems of Rigid Bodies Subject to Unilateral Constraints”. In: *SIAM Journal on Applied Mathematics* 42.2 (1982), pp. 281–296. ISSN: 00361399.

[137] B. Mirtich and J. Canny. “Impulse-based simulation of rigid bodies”. In: *Proceedings of the 1995 symposium on Interactive 3D Graphics*. ACM, 1995, p. 181. DOI: 10.1145/199404.199436.

[138] B. Mirtich. “Impulse-based dynamic simulation of rigid body systems”. PhD thesis. University of California at Berkeley, 1996.

[139] V. Krautler, W. F. van Gunsteren, and P. H. Hunenberger. “A fast SHAKE algorithm to solve distance constraint equations for small molecules in molecular dynamics simulations”. In: *Journal of Computational Chemistry* 22.5 (Apr. 2001), pp. 501–508. ISSN: 0192-8651. DOI: 10.1002/1096-987X(20010415)22:5<501::AID-JCC1021>3.0.CO;2-V.

[140] H. C. Andersen. “Rattle: A “velocity” version of the shake algorithm for molecular dynamics calculations”. In: *Journal of Computational Physics* 52.1 (Oct. 1983), pp. 24–34. ISSN: 00219991. DOI: 10.1016/0021-9991(83)90014-1.

[141] S. Miyamoto and P. A. Kollman. “Settle: An analytical version of the SHAKE and RATTLE algorithm for rigid water models”. In: *Journal of Computational Chemistry* 13.8 (Oct. 1992), pp. 952–962. ISSN: 0192-8651. DOI: 10.1002/jcc.540130805.

[142] P. Eastman and V. S. Pande. “Constant Constraint Matrix Approximation: A Robust, Parallelizable Constraint Method for Molecular Simulations”. In: *Journal of Chemical Theory and Computation* 6.2 (Feb. 2010), pp. 434–437. ISSN: 1549-9618. DOI: 10.1021/ct900463w.

[143] M. O’Connor. *BXD Impulse Constraint Example*. URL: https://gist.github.com/mikeoconnor0308/7e918daed7fb26c3317eb85f3fe1ec92 (visited on 11/29/2019).

[144] M. O’Connor. *Input files and dataset for CD3CN+F Reaction sampling in CHARMM*. June 2016. DOI: 10.5281/ZENODO.2276687.

[145] D. R. Glowacki, A. J. Orr-Ewing, and J. N. Harvey. “A parallel multistate framework for atomistic non-equilibrium reaction dynamics of solutes in strongly interacting organic solvents”. en. In: *ArXiv e-prints* 1412.3180 (Dec. 2014), p. 58. arXiv: 1412.4180.

[146] G. Díaz Leines and B. Ensing. “Path Finding on High-Dimensional Free Energy Landscapes”. In: *Physical Review Letters* 109.2 (July 2012), p. 020601. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.109.020601.

[147] B. Efron and R. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall, 1994, p. 436. ISBN: 0412042312.

[148] M. O’Connor et al. “Nano Simbox: An OpenCL-accelerated Framework for Interactive Molecular Dynamics”. In: *Proceedings of the 3rd International Workshop on OpenCL*. {IWOCL} ’15. ACM, 2015, 20:1–20:1. ISBN: 978-1-4503-3484-6. DOI: 10.1145/2791321.2791341.

[149] M. O’Connor et al. “Sampling molecular conformations and dynamics in a multiuser virtual reality framework”. In: *Science Advances* 4.6 (June 2018), eaat2731. ISSN: 2375-2548. DOI: 10.1126/sciadv.aat2731.

[150] M. O’Connor. *Input Files and Results for Benchmarking the iMD-VR Framework*. Dec. 2018. DOI: 10.5281/ZENODO.2352815.

[151] F. P. Brooks et al. “Project GROPEHaptic displays for scientific visualization”. In: *Proceedings of the 17th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’90*. Vol. 24. 4. New York, New York, USA: ACM Press, 1990, pp. 177–185. ISBN: 0201509334. DOI: 10.1145/97879.97899.

[152] M. C. Surles et al. “Sculpting proteins interactively: Continual energy minimization embedded in a graphical modeling system”. In: *Protein Science* 3.2 (Dec. 2008), pp. 198–210. ISSN: 09618368. DOI: 10.1002/pro.5560030205.

[153] J. E. Stone, J. Gullingsrud, and K. Schulten. “A system for interactive molecular dynamics simulation”. In: *Proceedings of the 2001 symposium on Interactive 3D graphics - SI3D ’01*. New York, New York, USA: ACM Press, 2001, pp. 191–194. ISBN: 1581132921. DOI: 10.1145/364338.364398.

[154] J. E. Stone et al. “GPU-accelerated molecular modeling coming of age”. In: *Journal of Molecular Graphics and Modelling* 29.2 (Sept. 2010), pp. 116–125. ISSN: 10933263. DOI: 10.1016/j.jmgm.2010.06.010.

[155] P. Grayson, E. Tajkhorshid, and K. Schulten. “Mechanisms of Selectivity in Channels and Enzymes Studied with Interactive Molecular Dynamics”. In: *Biophysical Journal* 85.1 (July 2003), pp. 36–48. ISSN: 00063495. DOI: 10.1016/S0006-3495(03)74452-X.

[156] J. Cohen and K. Schulten. “Mechanism of Anionic Conduction across ClC”. In: *Biophysical Journal* 86.2 (Feb. 2004), pp. 836–845. ISSN: 00063495. DOI: 10.1016/S0006-3495(04)74159-4.

[157] L. G. Trabuco et al. “The Role of L1 Stalk–tRNA Interaction in the Ribosome Elongation Cycle”. In: *Journal of Molecular Biology* 402.4 (Oct. 2010), pp. 741–760. ISSN: 00222836. DOI: 10.1016/j.jmb.2010.07.056.

[158] O. Delalande et al. “Complex molecular assemblies at hand via interactive simulations”. In: *Journal of Computational Chemistry* 30.15 (Nov. 2009), pp. 2375–2387. ISSN: 01928651. DOI: 10.1002/jcc.21235.

[159] M. Dreher et al. “Interactive Molecular Dynamics: Scaling up to Large Systems”. In: *Procedia Computer Science*. 2013 International Conference on Computational Science 18 (Jan. 2013), pp. 20–29. ISSN: 18770509. DOI: 10.1016/j.procs.2013.05.165.

[160] M. P. Haag and M. Reiher. “Real-time quantum chemistry”. In: *International Journal of Quantum Chemistry* 113.1 (Jan. 2013), pp. 8–20. ISSN: 00207608. DOI: 10.1002/qua.24336.

[161] A. C. Vaucher, M. P. Haag, and M. Reiher. “Real-time feedback from iterative electronic structure calculations”. In: *Journal of Computational Chemistry* 37.9 (Apr. 2016), pp. 805–812. ISSN: 01928651. DOI: 10.1002/jcc.24268.

[162] M. P. Haag and M. Reiher. “Studying chemical reactivity in a virtual environment”. In: *Faraday Discuss.* 169 (Oct. 2014), pp. 89–118. ISSN: 1359-6640. DOI: 10.1039/C4FD00021H.

[163] N. Luehr, A. G. B. Jin, and T. J. Martínez. “Ab Initio Interactive Molecular Dynamics on Graphical Processing Units (GPUs)”. In: *Journal of Chemical Theory and Computation* 11.10 (Oct. 2015), pp. 4536–4544. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.5b00419.

[164] S. Cooper et al. “Predicting protein structures with a multiplayer online game”. In: *Nature* 466.7307 (Aug. 2010), pp. 756–760. ISSN: 0028-0836. DOI: 10.1038/nature09304.

[165] F. Khatib et al. “Crystal structure of a monomeric retroviral protease solved by protein folding game players”. In: *Nature Structural & Molecular Biology* 18.10 (Oct. 2011), pp. 1175–1177. ISSN: 1545-9993. DOI: 10.1038/nsmb.2119.

[166] C. B. Eiben et al. “Increased Diels-Alderase activity through backbone remodeling guided by Foldit players”. In: *Nature Biotechnology* 30.2 (Feb. 2012), pp. 190–192. ISSN: 1087-0156. DOI: 10.1038/nbt.2109.

[167] F. Khatib et al. “Algorithm discovery by protein folding game players”. In: *Proceedings of the National Academy of Sciences* 108.47 (Nov. 2011), pp. 18949–18953. ISSN: 0027-8424. DOI: 10.1073/pnas.1115898108.

[168] C. Woods. “Interactive Visualisation and Data Analysis of Simulations using Jupyter Notebooks”. In: *HPC-Sig Remote Visualisation Workshop*. 2018.

[169] H. Jonsson, G. Mills, and K. W. Jacobsen. “Nudged elastic band method for finding minimum energy paths of transitions”. In: *Classical and Quantum Dynamics in Condensed Phase Simulations*. WORLD SCIENTIFIC, June 1998, pp. 385–404. ISBN: 978-981-02-3498-0. DOI: 10.1142/9789812839664_0016. arXiv: arXiv:1011.1669v3.

[170] G. Henkelman and H. Jónsson. “Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points”. In: *The Journal of Chemical Physics* 113.22 (Dec. 2000), pp. 9978–9985. ISSN: 0021-9606. DOI: 10.1063/1.1323224.

[171] D. J. Wales and J. M. Carr. “Quasi-Continuous Interpolation Scheme for Pathways between Distant Configurations”. In: *Journal of Chemical Theory and Computation* 8.12 (Dec. 2012), pp. 5020–5034. ISSN: 1549-9618. DOI: 10.1021/ct3004832.

[172] D. Swapp, V. Pawar, and C. Loscos. “Interaction with co-located haptic feedback in virtual reality”. In: *Virtual Reality* 10.1 (May 2006), pp. 24–30. ISSN: 1359-4338. DOI: 10.1007/s10055-006-0027-5.

[173] F. Brooks. “What’s real about virtual reality?” In: *IEEE Computer Graphics and Applications* 19.6 (1999), pp. 16–27. ISSN: 02721716. DOI: 10.1109/38.799723.

[174] I. E. Sutherland. “A head-mounted three dimensional display”. In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS ’68 (Fall, part I)*. New York, New York, USA: ACM Press, 1968, p. 757. ISBN: 158113052X. DOI: 10.1145/1476589.1476686.

[175] I. E. Sutherland. “The Ultimate Display”. In: *Proceedings of the IFIP Congress* 2 (1965), pp. 506–508.

[176] M. Slater and M. V. Sanchez-Vives. “Enhancing Our Lives with Immersive Virtual Reality”. In: *Frontiers in Robotics and AI* 3 (Dec. 2016), p. 74. ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00074.

[177] M. Slater. “Grand Challenges in Virtual Environments”. In: *Frontiers in Robotics and AI* 1 (May 2014), p. 3. ISSN: 2296-9144. DOI: 10.3389/frobt.2014.00003.

[178] N. E. Seymour et al. “Virtual Reality Training Improves Operating Room Performance”. In: *Annals of Surgery* 236.4 (Oct. 2002), pp. 458–464. ISSN: 0003-4932. DOI: 10.1097/00000658-200210000-00008.

[179] Valve. *SteamVR Tracking*. 2018.

[180] J. Lanier. *Dawn of the New Everything: A Journey Through Virtual Reality*. The Bodley Head, 2017. ISBN: 1627794093.

[181] M. J. Schuemie et al. “Research on Presence in Virtual Reality: A Survey”. In: *CyberPsychology & Behavior* 4.2 (Apr. 2001), pp. 183–201. ISSN: 1094-9313. DOI: 10.1089/109493101300117884.

[182] R. T. Azuma. “The Most Important Challenge Facing Augmented Reality”. In: *Presence: Teleoperators and Virtual Environments* 25.3 (Dec. 2016), pp. 234–238. ISSN: 1054-7460. DOI: 10.1162/PRES_a_00264.

[183] T. D. Goddard et al. “Molecular Visualization on the Holodeck”. In: *Journal of Molecular Biology* 430.21 (Oct. 2018), pp. 3982–3996. ISSN: 00222836. DOI: 10.1016/j.jmb.2018.06.040.

[184] M. Norrby et al. “Molecular Rift: Virtual Reality for Drug Designers”. In: *Journal of Chemical Information and Modeling* 55.11 (Nov. 2015), pp. 2475–2484. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.5b00544.

[185] E. M. Ratamero et al. “Touching proteins with virtual bare hands”. In: *Journal of Computer-Aided Molecular Design* 32.6 (June 2018), pp. 703–709. ISSN: 0920-654X. DOI: 10.1007/s10822-018-0123-0. arXiv: 1710.03655.

[186] S. Doutreligne et al. “UnityMol: interactive and ludic visual manipulation of coarse-grained RNA and other biomolecules”. In: *2015 IEEE 1st International Workshop on Virtual and Augmented Reality for Molecular Science (VARMS@IEEEVR)*. IEEE, Mar. 2015, pp. 1–6. ISBN: 978-1-4673-6926-8. DOI: 10.1109/VARMS.2015.7151718.

[187] M. WRIGHT. “Open Sound Control: an enabling technology for musical networking”. In: *Organised Sound* 10.03 (Nov. 2005), p. 193. ISSN: 1355-7718. DOI: 10.1017/S1355771805000932.

[188] N. Allinger, Y. Yuh, and J. Lii. “Molecular mechanics. The MM3 force field for hydrocarbons. 1”. In: *Journal of the American Chemical ...* 111.23 (1989), pp. 8551–8566.

[189] J. H. Lii and N. L. Allinger. “Molecular mechanics. The MM3 force field for hydrocarbons. 2. Vibrational frequencies and thermodynamics”. In: *Journal of the American Chemical Society* 111.23 (Nov. 1989), pp. 8566–8575. ISSN: 0002-7863. DOI: 10.1021/ja00205a002.

[190] J. H. Lii and N. L. Allinger. “Molecular mechanics. The MM3 force field for hydrocarbons. 3. The van der Waals’ potentials and crystal data for aliphatic and aromatic hydrocarbons”. In: *Journal of the American Chemical Society* 111.23 (Nov. 1989), pp. 8576–8582. ISSN: 0002-7863. DOI: 10.1021/ja00205a003.

[191] P. Eastman et al. “OpenMM 7: Rapid development of high performance algorithms for molecular dynamics”. In: *PLOS Computational Biology* 13.7 (July 2017). Ed. by R. Gentleman, e1005659. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005659.

[192] Valve. *SteamVR Tracking*. 2018. URL: https://partner.steamgames.com/vrlicensing (visited on 11/29/2019).

[193] J. E. Stone et al. "Immersive Molecular Visualization and Interactive Modeling with Commodity Hardware". en. In: *International Symposium on Visual Computing*. Springer Berlin Heidelberg, Nov. 2010, pp. 382–393. DOI: 10.1007/978-3-642-17274-8_38.

[194] V. Thiruselvam et al. "Crystal structure analysis of a hypothetical protein (MJ0366) from Methanocaldococcus jannaschii revealed a novel topological arrangement of the knot fold". In: *Biochemical and Biophysical Research Communications* 482.2 (Jan. 2017), pp. 264–269. ISSN: 0006291X. DOI: 10.1016/j.bbrc.2016.11.052.

[195] C. Lewis and J. Rieman. "Creating the Initial Design". In: *Task-centered user interface design: A Practical Introduction*. 1993, pp. 27–38.

[196] J. W. McGowan and T. Janiszewski. *Pinch-and-zoom, zoom-and-pinch gesture control*. Mar. 2014.

[197] S. Machkovech. "Learning how to VR with Tilt Brush, HTC Vive's killer app". In: *Ars Technica* 5 (2016).

[198] T. Page. "Skeuomorphism or flat design: future directions in mobile device User Interface (UI) design education". In: *International Journal of Mobile Learning and Organisation* 8.2 (2014), p. 130. ISSN: 1746-725X. DOI: 10.1504/IJMLO.2014.062350.

[199] C. J. Woods et al. "A water-swap reaction coordinate for the calculation of absolute protein-ligand binding free energies". In: *Journal of Chemical Physics* 134.5 (Feb. 2011), p. 054114. ISSN: 00219606. DOI: 10.1063/1.3519057.

[200] C. Lewis and J. Rieman. "Testing The Design With Users". In: *Task-centered user interface design: A Practical Introduction*. 1993, pp. 79–97.

[201] M. Hitchens. "A Survey of First-person Shooters and their Avatars". In: *Game Studies*. Vol. 11. 3. 2011, pp. 96–120. DOI: 10.1145/1746050.1746054.

[202] C. J. Woods et al. "Computational Assay of H7N9 Influenza Neuraminidase Reveals R292K Mutation Reduces Drug Binding Affinity". In: *Scientific Reports* 3.1 (Dec. 2013), p. 3561. ISSN: 2045-2322. DOI: 10.1038/srep03561.

[203] L. Brocchieri. “Protein length in eukaryotic and prokaryotic proteomes”. In: *Nucleic Acids Research* 33.10 (June 2005), pp. 3390–3400. ISSN: 0305-1048. DOI: 10.1093/nar/gki615.

[204] M.-P. Lefranc et al. “IMGT®, the international ImMunoGeneTics information system® 25 years on”. In: *Nucleic Acids Research* 43.D1 (Jan. 2015), pp. D413–D422. ISSN: 1362-4962. DOI: 10.1093/nar/gku1056.

[205] D. Beeler, E. Hutchins, and P. Pedriana. *Asynchronous spacewarp*. 2016. URL: https://developer.oculus.com/blog/asynchronous-spacewarp (visited on 11/29/2019).

[206] M. Claypool, K. Claypool, and F. Damaa. “The effects of frame rate and resolution on users playing first person shooter games”. In: *Multimedia Computing and Networking 2006*. Ed. by S. Chandra and C. Griwodz. Vol. 6071. International Society for Optics and Photonics, Jan. 2006, p. 607101. DOI: 10.1117/12.648609.

[207] Unity3D. *What's new in Unity 2018.1 - Unity*. URL: https://unity3d.com/unity/whats-new/unity-2018.1.0 (visited on 09/07/2018).

[208] A. Munshi. “The OpenCL specification”. In: *2009 IEEE Hot Chips 21 Symposium (HCS)*. IEEE, Aug. 2009, pp. 1–314. ISBN: 978-1-4673-8873-3. DOI: 10.1109/HOTCHIPS.2009.7478342.

[209] B. Kozlíková et al. “Visualization of Biomolecular Structures: State of the Art Revisited”. In: *Computer Graphics Forum* 36.8 (Dec. 2017), pp. 178–204. ISSN: 01677055. DOI: 10.1111/cgf.13072.

[210] D. Luebke et al. *Level of Detail for 3D Graphics : Application and Theory*. 2002, p. 431. ISBN: 9780080510118.

[211] K. Goh and Y. Chen. “Controlling water transport in carbon nanotubes”. In: *Nano Today* 14 (June 2017), pp. 13–15. ISSN: 17480132. DOI: 10.1016/j.nantod.2016.12.015. arXiv: arXiv:1011.1669v3.

[212] W. R. Taylor and K. Lin. “Protein knots: A tangled problem”. In: *Nature* 421.6918 (Jan. 2003), p. 25. ISSN: 0028-0836. DOI: 10.1038/421025a.

[213] B. Derrick, D. Toher, and P. White. “Why Welch's test is Type I error robust”. In: *The Quantitative Methods in Psychology* 12.1 (2016), pp. 30–38. ISSN: 2292-1354.

[214] E. Jones, T. Oliphant, and P. Peterson. *SciPy: open source scientific tools for Python*. 2014.

[215] A. C. Vaucher and M. Reiher. “Minimum Energy Paths and Transition States by Curve Optimization”. In: *Journal of Chemical Theory and Computation* 14.6 (June 2018), pp. 3091–3099. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.8b00169. arXiv: 1802.05669.

[216] D. Branduardi and J. D. Faraldo-Gómez. “String Method for Calculation of Minimum Free-Energy Paths in Cartesian Space in Freely Tumbling Systems”. In: *Journal of Chemical Theory and Computation* 9.9 (Sept. 2013), pp. 4140–4154. ISSN: 1549-9618. DOI: 10.1021/ct400469w.

[217] L.-P. Wang et al. “Automated Discovery and Refinement of Reactive Molecular Dynamics Pathways”. In: *Journal of Chemical Theory and Computation* 12.2 (Feb. 2016), pp. 638–649. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.5b00830.

[218] M. O’Connor. *Input files and data for path generation of alanine dipeptide isomerization in virtual reality*. Dec. 2018. DOI: 10.5281/ZENODO.2385199.

[219] J. Apostolakis, P. Ferrara, and A. Caflisch. “Calculation of conformational transitions and barriers in solvated systems: Application to the alanine dipeptide in water”. In: *The Journal of Chemical Physics* 110.4 (Jan. 1999), pp. 2099–2108. ISSN: 0021-9606. DOI: 10.1063/1.477819.

[220] P. G. Bolhuis, C. Dellago, and D. Chandler. “Reaction coordinates of biomolecular isomerization”. In: *Proceedings of the National Academy of Sciences* 97.11 (May 2000), pp. 5877–5882. ISSN: 0027-8424. DOI: 10.1073/pnas.100127697.

[221] L. Maragliano et al. “String method in collective variables: Minimum free energy paths and isocommittor surfaces”. In: *The Journal of Chemical Physics* 125.2 (July 2006), p. 024106. ISSN: 0021-9606. DOI: 10.1063/1.2212942.

[222] T. Lazaridis et al. “Reaction paths and free energy profiles for conformational transitions: An internal coordinate approach”. In: *The Journal of Chemical Physics* 95.10 (Nov. 1991), pp. 7612–7625. ISSN: 0021-9606. DOI: 10.1063/1.461335.

[223] R. Crehuet and M. J. Field. “A temperature-dependent nudged-elastic-band algorithm”. In: *The Journal of Chemical Physics* 118.21 (June 2003), pp. 9563–9571. ISSN: 0021-9606. DOI: 10.1063/1.1571817.

[224] A. Ma and A. R. Dinner. “Automatic Method for Identifying Reaction Coordinates in Complex Systems”. In: *The Journal of Physical Chemistry B* 109.14 (2005), pp. 6769–6779. ISSN: 1520-6106. DOI: 10.1021/jp045546c.

[225] W. Ren et al. “Transition pathways in complex systems: Application of the finite-temperature string method to the alanine dipeptide”. In: *The Journal of Chemical Physics* 123.13 (Oct. 2005), p. 134109. ISSN: 0021-9606. DOI: 10.1063/1.2013256.

[226] S. Bahn and K. Jacobsen. “An object-oriented scripting interface to a legacy electronic structure code”. In: *Computing in Science & Engineering* 4.3 (2002), pp. 56–66. ISSN: 15219615. DOI: 10.1109/5992.998641.

[227] E. Formoso, V. Limongelli, and M. Parrinello. “Energetics and Structural Characterization of the large-scale Functional Motion of Adenylate Kinase”. In: *Scientific Reports* 5.1 (July 2015), p. 8425. ISSN: 2045-2322. DOI: 10.1038/srep08425.

[228] PLUMED. *Belfast tutorial: Adaptive variables 1*. URL: https://plumed.github.io/doc-v2.5/user-doc/html/belfast-2.html (visited on 01/08/2019).

[229] D. Branduardi, G. Bussi, and M. Parrinello. “Metadynamics with Adaptive Gaussians”. In: *Journal of Chemical Theory and Computation* 8.7 (July 2012), pp. 2247–2254. ISSN: 1549-9618. DOI: 10.1021/ct3002464. arXiv: 1205.4300.

[230] G. Hu et al. “Deep Multi-Task Learning to Recognise Subtle Facial Expressions of Mental States”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 103–119.

[231] P. Tao et al. “Comparison of Three Chain-of-States Methods: Nudged Elastic Band and Replica Path with Restraints or Constraints”. In: *Journal of Chemical Theory and Computation* 8.12 (Dec. 2012), pp. 5035–5051. ISSN: 1549-9618. DOI: 10.1021/ct3006248.

[232] B. Peters et al. “A growing string method for determining transition states: Comparison to the nudged elastic band and string methods”. In: *Journal of Chemical Physics* 120.17 (May 2004), pp. 7877–7886. ISSN: 00219606. DOI: 10.1063/1.1691018.

[233] C. Bergonzo, C. Simmerling, and R. Walker. *Amber Advanced Workshop - Tutorial 5 - Nudged Elastic Band*.

[234] M. L. Mansfield. “Are there knots in proteins?” In: *Nature Structural & Molecular Biology* 1.4 (Apr. 1994), pp. 213–214. ISSN: 1545-9993. DOI: 10.1038/nsb0494-213.

[235] N. C. H. Lim and S. E. Jackson. “Molecular knots in biology and chemistry”. In: *Journal of Physics: Condensed Matter* 27.35 (Sept. 2015), p. 354101. ISSN: 0953-8984. DOI: 10.1088/0953-8984/27/35/354101.

[236] W. R. Taylor. “A deeply knotted protein structure and how it might fold”. In: *Nature* 406.6798 (Aug. 2000), pp. 916–919. ISSN: 0028-0836. DOI: 10.1038/35022623.

[237] A. L. Mallam. “How does a knotted protein fold?” In: *FEBS Journal* 276.2 (Jan. 2009), pp. 365–375. ISSN: 1742464X. DOI: 10.1111/j.1742-4658.2008.06801.x.

[238] F. Ziegler et al. “Knotting and unknotting of a protein in single molecule experiments”. In: *Proceedings of the National Academy of Sciences* 113.27 (July 2016), pp. 7533–7538. ISSN: 0027-8424. DOI: 10.1073/pnas.1600614113.

[239] S.-C. Lou et al. “The Knotted Protein UCH-L1 Exhibits Partially Unfolded Forms under Native Conditions that Share Common Structural Features with Its Kinetic Folding Intermediates”. In: *Journal of Molecular Biology* 428.11 (June 2016), pp. 2507–2520. ISSN: 00222836. DOI: 10.1016/j.jmb.2016.04.002.

[240] J. K. Noel, J. I. Sulkowska, and J. N. Onuchic. “Slipknotting upon native-like loop formation in a trefoil knot protein”. In: *Proceedings of the National Academy of Sciences* 107.35 (Aug. 2010), pp. 15403–15408. ISSN: 0027-8424. DOI: 10.1073/pnas.1009522107.

[241] S. Wallin, K. B. Zeldovich, and E. I. Shakhnovich. “The Folding Mechanics of a Knotted Protein”. In: *Journal of Molecular Biology* 368.3 (May 2007), pp. 884–893. ISSN: 00222836. DOI: 10.1016/j.jmb.2007.02.035.

[242] S. Doerr and G. De Fabritiis. “On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations”. In: *Journal of Chemical Theory and Computation* 10.5 (May 2014), pp. 2064–2069. ISSN: 1549-9618. DOI: 10.1021/ct400919u.

[243] X. Huang et al. “Rapid equilibrium sampling initiated from nonequilibrium data”. In: *Proceedings of the National Academy of Sciences* 106.47 (Nov. 2009), pp. 19765–19769. ISSN: 0027-8424. DOI: 10.1073/pnas.0909088106.

[244] E. Z. Eisenmesser et al. “Intrinsic dynamics of an enzyme underlies catalysis”. In: *Nature* 438.7064 (Nov. 2005), pp. 117–121. ISSN: 0028-0836. DOI: 10.1038/nature04105.

[245] C. N. Chi et al. “A Structural Ensemble for the Enzyme Cyclophilin Reveals an Orchestrated Mode of Action at Atomic Resolution”. In: *Angewandte Chemie (International ed. in English)* 54.40 (2015), pp. 11657–11661. DOI: 10.1002/anie.201503698.

[246] D. R. Glowacki, J. N. Harvey, and A. J. Mulholland. “Protein dynamics and enzyme catalysis: the ghost in the machine?” In: *Biochemical Society Transactions* 40 (2012), pp. 515–521. ISSN: 0300-5127. DOI: 10.1042/BST20120047.

[247] T. R. Gamble et al. “Crystal structure of human cyclophilin A bound to the amino-terminal domain of HIV-1 capsid”. In: *Cell* 87.7 (1996), pp. 1285–1294. ISSN: 00928674. DOI: 10.1016/S0092-8674(00)81823-1.

[248] R. B. Best, G. Hummer, and W. a. Eaton. “Native contacts determine protein folding mechanisms in atomistic simulations”. In: *Proceedings of the National Academy of Sciences* 110.44 (Oct. 2013), pp. 17874–17879. ISSN: 0027-8424. DOI: 10.1073/pnas.1311599110.

[249] Y. Lu et al. “Feature selection using principal feature analysis”. In: *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA ’07*. New York, New York, USA: ACM Press, 2007, p. 301. ISBN: 9781595937025. DOI: 10.1145/1291233.1291297.

[250] M. I. Zimmerman and G. R. Bowman. “FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs”. In: *Journal of Chemical Theory and Computation* 11.12 (Dec. 2015), pp. 5747–5757. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.5b00737.

[251] R. Zhou. “Free energy landscape of protein folding in water: Explicit vs. implicit solvent”. In: *Proteins: Structure, Function, and Genetics* 53.2 (Nov. 2003), pp. 148–161. ISSN: 0887-3585. DOI: 10.1002/prot.10483.

[252] B. Zagrovic, E. J. Sorin, and V. Pande. “β-hairpin folding simulations in atomistic detail using an implicit solvent model 1 1Edited by F. Cohen”. In: *Journal of Molecular Biology* 313.1 (Oct. 2001), pp. 151–169. ISSN: 00222836. DOI: 10.1006/jmbi.2001.5033.

[253] I. R. Kleckner and M. P. Foster. “An introduction to NMR-based approaches for measuring protein dynamics”. In: *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* 1814.8 (Aug. 2011), pp. 942–968. ISSN: 15709639. DOI: 10.1016/j.bbapap.2010.10.012.

[254] A. Grossfield and D. M. Zuckerman. “Quantifying uncertainty and sampling quality in biomolecular simulations.” In: *Annual reports in computational chemistry* 5 (Jan. 2009), pp. 23–48. ISSN: 1574-1400. DOI: 10.1016/S1574-1400(09)00502-7.

[255] H. H. Ku. “Notes on the use of propagation of error formulas”. In: *Journal of Research of the National Bureau of Standards* 70.4 (Oct. 1966), p. 263. ISSN: 0022-4316. DOI: 10.6028/jres.070C.025.